

**UNIVERSIDAD AUTÓNOMA DE MADRID**

**ESCUELA POLITÉCNICA SUPERIOR**



## **TRABAJO FIN DE GRADO**

### **DESARROLLO DE UNA APLICACIÓN ANDROID PARA EL CONTROL AUTOMÁTICO DE LA ASISTENCIA A CLASE**

**Cristian Eric Grange García**

**Julio 2014**

**Tutor:**  
**Luis Fernando Lago Fernández**



# Resumen

---

Actualmente, los profesores en universidades y colegios verifican la asistencia de los estudiantes manualmente, bien llamando a cada alumno por su nombre o bien pasando una hoja de firmas en cada clase [1]. El uso de estos métodos es incómodo y consume mucho tiempo, además de consumir demasiado papel de forma innecesaria. La aplicación presentada en este proyecto nace de la necesidad de mejorar este proceso y reducir el tiempo y los materiales empleados para anotar la asistencia de los estudiantes.

En este proyecto se ha desarrollado AppTendance, una aplicación para Android para verificar la asistencia usando el teléfono móvil del profesor de forma rápida, precisa, y sin necesidad de gastar papel. Ésta aplicación instalada en el teléfono o tableta del profesor permite conectar con los teléfonos o tabletas de los alumnos por medio de Bluetooth y, mediante la transmisión de la dirección “Media Access Control” (MAC) del dispositivo de cada alumno al dispositivo del profesor, confirmar su presencia. De esta forma, el dispositivo de cada alumno queda registrado en la base de datos del dispositivo del profesor, evitando suplantaciones de identidad durante el proceso. Además, dado que el protocolo de comunicaciones inalámbricas Bluetooth no asegura la presencia del dispositivo remoto en un rango limitado exacto, se hace necesario que el profesor facilite un código de control, una contraseña, que sólo conocerán los alumnos presentes en el aula. Para facilitar la tarea de crear las asignaturas en la aplicación se permite cargar la información de los alumnos en la aplicación desde ficheros en formato CSV almacenados en la memoria externa del dispositivo, así como generar automáticamente detallados informes de asistencia para ser imprimidos o exportados a otras plataformas, facilitando así la organización del material recolectado y agilizando las consultas.

La aplicación presentada en esta memoria ofrece la posibilidad de intercambiar información entre el profesor y sus alumnos a través de su dispositivo móvil Android, y, de esta forma, permitir clases más dinámicas gracias al *feedback* proporcionado al profesor en tiempo real y de manera anónima. Aprovechando esta posibilidad, además de controlar la asistencia, el alumno puede enviar preguntas sobre la asignatura al profesor mientras éste imparte la lección, de forma que los demás alumnos no sepan quién formuló la pregunta y perder así el miedo a ser juzgado por preguntar.

## Palabras clave

Bluetooth, Asistencia, Dispositivos Móviles, Android, Teléfonos Inteligentes, Comunicación Inalámbrica, Ayuda al Profesor, Media Access Control, Control de Asistencia.



# Abstract

---

Nowadays teachers from schools and universities verify the attendance of their students manually calling each student by name or passing a sheet of signatures in each class [1]. Using these methods is uncomfortable, time consuming, and wastes a lot of paper unnecessarily. The application presented in this document stems from the need to improve the process and reduce the time and materials used to record the attendance of students.

This document therefore presents the development of AppTendance, an Android application to verify the attendance using mobile devices of both teacher and students, quickly, accurately and without wasting paper. An application installed on the teacher's Android phone or tablet connects with the student's mobile devices via Bluetooth and confirms their attendance by forwarding each device's "*Media Access Control*" (MAC) identification to the teacher. Thus, each student device is registered in the database of teacher's device, avoiding identity theft during the process. Furthermore, since Bluetooth wireless communication protocol does not ensure the presence of the remote device at an exact limited range, it is necessary that the teacher facilitates a control code, a password that will only be known by the students present in the classroom. In order to facilitate the task of creating subjects in the application, it is possible to load the students' information into the application from CSV format files stored in the device's external memory, and also automatically generate detailed attendance reports to be printed or exported to other platforms, thus facilitating the organization of the collected material and speeding up queries.

The application presented herein provides the possibility to exchange information between teacher and students through its Android mobile device, and thus, allow for more dynamic classes through the feedback provided to the teacher in real time and anonymously. Taking this opportunity, in addition to monitor attendance, students can send questions to the teacher about the subject as he is teaching, so that the other students won't be able to know who asked, losing the fear of being judged.

## Keywords

Bluetooth, Attendance, Mobile Devices, Android, Smartphones, Wireless Communication, Teacher Support, Media Access Control, Attendance Control.



# Índice

---

Palabras clave .....	3
Keywords.....	5

## 1. Introducción

1.1. Motivación.....	17
1.1. Objetivos.....	18
1.2. Estructura de la memoria.....	18

## 2. Estado del Arte

2.1. Descripción de sistemas existentes.....	20
2.1.1. Tecnologías para registrar la asistencia del estudiante.....	20
2.1.1.1. Pasar lista de forma manual.....	20
2.1.1.2. Asociación de datos infrarrojos (IrDA) .....	21
2.1.1.3. Identificación por radiofrecuencia (RFID) .....	21
2.1.1.4. Comunicación de campo cercano (NFC).....	21
2.1.1.5. Biometría .....	22
➤ Reconocimiento ocular o dactilar.....	22
➤ Reconocimiento facial.....	22
2.1.1.6 GPS-GPRS .....	23
2.1.1.7 Bluetooth .....	23
2.1.2. Tecnologías para almacenar la información.....	24
2.1.2.1. Plataforma web .....	24
2.1.2.2. Base de datos SQLite.....	24
2.2. Aportaciones del sistema desarrollado .....	25
• Ahorro de tiempo y recursos .....	25
• No necesita conexión a Internet .....	25
• No necesita mantenimiento .....	25
• No necesita inversión inicial.....	26
• Permite el intercambio de información .....	26
• Proporciona confirmación del procedimiento .....	26
• Funcionamiento alternativo .....	26

### 3. Diseño y desarrollo

3.1.	Tecnologías utilizadas .....	27
3.1.1.	Comunicación inalámbrica: Bluetooth.....	27
3.1.2.	Sistema operativo: Android.....	28
3.2.	Entorno de desarrollo.....	32
3.3.	Arquitectura del sistema .....	33
3.3.1.	Módulo de actividades del profesor .....	34
➤	Editar, añadir y borrar asignaturas. ....	34
➤	Editar, añadir y borrar alumnos.....	35
➤	Registrar MAC de alumno .....	36
➤	Pasar lista.....	37
➤	Consultar listas de asistencia.....	39
➤	Recibir preguntas de los estudiantes .....	40
3.3.2.	Módulo de actividades del alumno.....	41
➤	Información personal.....	41
➤	Agregar, editar o eliminar asignaturas .....	42
➤	Confirmar asistencia.....	43
➤	Consultar asistencia propia.....	44
➤	Enviar preguntas al profesor .....	44
3.3.3.	Módulo de conexiones Bluetooth.....	45
3.3.4.	Módulo de manejo de base de datos.....	46
3.3.5.	Módulo de manejo de ficheros CSV .....	46
3.4.	Descripción de la base de datos.....	47
3.4.1.	Tabla de alumnos .....	48
3.4.2.	Tabla de asignaturas creadas .....	48
3.4.3.	Tabla de asignaturas registradas.....	48
3.4.4.	Tabla de alumnos registrados.....	49
3.4.5.	Tabla de asistencia .....	49
3.4.6.	Tabla de asistencia propia .....	49
3.5.	Descripción de las conexiones Bluetooth.....	49

### 4. Evaluación del sistema

4.1.	Pruebas de uso .....	53
4.1.1.	Pruebas de navegación .....	53
4.1.2.	Pruebas de la base de datos .....	53
•	Crear / Editar / Borrar alumnos de una asignatura.....	53



• Crear / Editar / Borrar alumnos de la lista general de alumnos registrados.....	54
• Crear / Editar / Borrar asignaturas .....	54
• Cargar datos de un fichero CSV .....	54
• Anotar asistencia de forma manual.....	54
• Comprobar las listas de asistencia con todos los filtros.....	54
• Modificar las listas de asistencia .....	54
4.1.3. Pruebas de importación y exportación de ficheros CSV.....	55
4.1.4. Pruebas de las conexiones Bluetooth .....	55
4.2. Caso de uso general .....	55
4.2.1. Profesor .....	56
4.2.1.1. Crear asignatura.....	56
4.2.1.2. Registrar dispositivos de los alumnos .....	57
4.2.1.3. Pasar lista.....	58
4.2.1.4. Recibir mensajes.....	60
4.2.1.5. Consultar listas de asistencia .....	61
4.2.2. Alumno.....	64
4.2.2.1. Emparejar dispositivo .....	64
4.2.2.2. Registrar dispositivo .....	65
4.2.2.3. Confirmar asistencia .....	66
4.2.2.4. Enviar mensajes al profesor.....	67
4.3. Pruebas en distintos modelos y versiones de Android .....	67
4.4. Pruebas con usuarios .....	68
 5. Conclusiones y trabajo futuro .....	 69
 6. Referencias .....	 72
 7. Anexos	
7.1. Fichero CSV de ejemplo .....	76

# Índice de figuras

---

<b>Figura 1:</b> Esquema del funcionamiento básico de AppTendance.	27
<b>Figura 2:</b> Cuota de mercado de Sistemas Operativos en España. Kantar Worldpanel.	29
<b>Figura 3:</b> Distribución de la cuota de mercado de Android según versiones. Android Developers.	31
<b>Figura 4:</b> Simulador GenyMotion emulando un Samsung Galaxy S2.	33
<b>Figura 5:</b> Esquema de la organización de los módulos de AppTendance.	33
<b>Figura 6:</b> Pantalla de opciones del profesor.	34
<b>Figura 7:</b> Distintas pantallas de opciones para crear, editar o eliminar asignaturas del dispositivo del profesor.	35
<b>Figura 8:</b> Distintas pantallas de opciones para crear, editar o eliminar alumnos del dispositivo del profesor.	36
<b>Figura 9:</b> Pantalla de registro de alumnos y mensaje de confirmación del registro.	37
<b>Figura 10:</b> Distintas pantallas para pasar lista: de forma manual y mediante Bluetooth.	38
<b>Figura 11:</b> Pantalla de opciones para mostrar las listas de asistencia y lista de asistencia de una asignatura.	39
<b>Figura 12:</b> Pantalla de selección de fecha para consultar listas de asistencia, y resultados de búsqueda en lista de asistencia por fecha y por alumno.	40
<b>Figura 13:</b> Pantalla de recepción de mensajes del profesor y alerta de nuevo mensaje.	40
<b>Figura 14:</b> Pantalla de opciones del alumno.	41
<b>Figura 15:</b> Pantalla de edición de datos personales del alumno.	41
<b>Figura 16:</b> Proceso de búsqueda y emparejamiento del dispositivo del profesor de la asignatura y adición de ésta a la base de datos.	42
<b>Figura 17:</b> Confirmación para borrar una asignatura del dispositivo del alumno.	43

<b>Figura 18:</b> Pantallas de confirmación de la asistencia del estudiante.	44
<b>Figura 19:</b> Lista de asistencia del propio alumno a una asignatura.	44
<b>Figura 20:</b> Pantalla para enviar un mensaje al profesor.	45
<b>Figura 21:</b> Petición para activar Bluetooth en el dispositivo y mensaje de error en caso de que el usuario no dé el permiso.	50
<b>Figura 22:</b> Solicitud de asociación por Bluetooth con otro dispositivo.	51
<b>Figura 23:</b> Caso de uso en el que el usuario crea una asignatura y carga los datos desde un fichero CSV.	56
<b>Figura 24:</b> Caso de uso en el que el usuario registra los dispositivos de los alumnos.	57
<b>Figura 25:</b> Solicitud de permiso de Bluetooth para hacer el dispositivo visible.	57
<b>Figura 26:</b> Mensajes de confirmación y error tras registrar un alumno.	58
<b>Figura 27:</b> Caso de uso en el que el usuario pasa lista a su clase de forma manual.	59
<b>Figura 28:</b> Caso de uso en el que el usuario pasa lista a su clase mediante Bluetooth.	60
<b>Figura 29:</b> Recepción de mensajes de los alumnos.	61
<b>Figura 30:</b> Opciones de búsqueda en listas de asistencia.	61
<b>Figura 31:</b> Informe de asistencia total de la asignatura.	62
<b>Figura 32:</b> Lista de asistencia de la asignatura en una fecha concreta.	63
<b>Figura 33:</b> Lista de asistencia de un alumno concreto de la asignatura.	63
<b>Figura 34:</b> Caso de uso en el que el usuario empareja el dispositivo del profesor.	64
<b>Figura 35:</b> Caso de uso en el que el usuario registra su dispositivo añadiendo una nueva asignatura.	65
<b>Figura 36:</b> Caso de uso en el que el usuario confirma su asistencia a clase, y los	66

mensajes de confirmación o error recibidos tras realizar el procedimiento.

**Figura 37:** Caso de uso en el que el alumno envía un mensaje al profesor.

67

# Índice de tablas

---

<b>Tabla 1:</b> Cuota de mercado de los distintos sistemas operativos para Smartphone a nivel mundial. Kantar Worldpanel, Abril 2014.	30
<b>Tabla 2:</b> Cuota de mercado de las distintas versiones y niveles de API de Android. Android Developers.	31
<b>Tabla 3:</b> Detalle de las tablas que componen la base de datos de la aplicación.	48
<b>Tabla 4:</b> Modelos de dispositivos móviles Android físicos en los que se ha probado la aplicación.	67
<b>Tabla 5:</b> Modelos de dispositivos móviles Android simulados con GenyMotion.	68

# Glosario

---

**CSV.** Los ficheros CSV (del inglés comma-separated values) son un tipo de documento en formato abierto sencillo para representar datos en forma de tabla, en las que las columnas se separan por comas (o punto y coma en donde la coma es el separador decimal: España, Francia, Italia...) y las filas por saltos de línea. Los campos que contengan una coma, un salto de línea o una comilla doble deben ser encerrados entre comillas dobles.

**Actividad.** Una actividad es una única cosa concreta que el usuario puede hacer en una aplicación Android. Casi todas las actividades interaccionan con el usuario, por lo que las Actividades se encargan de crear una ventana en la que poner la interfaz de usuario con “`setContentView (View)`”. Mientras las actividades son muchas veces presentadas al usuario como ventanas a pantalla completa, también pueden ser utilizadas de otras formas como ventanas flotantes, o embebidas dentro de otra actividad. Todas las actividades deben estar declaradas en el fichero de Manifiesto de la aplicación.

**Layout.** Un *layout* define la estructura visual de una interfaz de usuario como la de una actividad o aplicación. Se puede declarar un layout mediante un fichero en formato XML, o bien instanciando los elementos en tiempo de ejecución de manera programática. La ventaja de declarar la interfaz de usuario en XML es que permite separar la presentación de la aplicación del código que controla su comportamiento.

**MAC.** En las redes de computadoras, la dirección MAC (siglas en inglés de media access control; en español "control de acceso al medio") es un identificador de 48 bits (6 bloques hexadecimales) que corresponde de forma única a una tarjeta o dispositivo de red. Se conoce también como dirección física, y es única para cada dispositivo. Está determinada y configurada por el IEEE (los últimos 24 bits) y el fabricante (los primeros 24 bits) utilizando el *organizationally unique* identifier. Las direcciones MAC son únicas a nivel mundial, puesto que son escritas directamente, en forma binaria, en el hardware en su momento de fabricación.

**Socket.** Socket designa un concepto abstracto por el cual dos programas (posiblemente situados en computadoras o dispositivos móviles distintos) pueden intercambiar cualquier flujo de datos, generalmente de manera fiable y ordenada.

**Piconet.** Es una red informática cuyos nodos se conectan utilizando Bluetooth. Una piconet puede constar de dos a ocho dispositivos, habiendo siempre un “maestro”, mientras que los demás son esclavos.

**Hilo de ejecución.** Un hilo es una unidad de ejecución concurrente. Tiene su propia pila de llamadas a métodos invocados, sus argumentos y sus variables locales. Cada aplicación tiene al menos un hilo ejecutándose cuando comienza, el hilo principal. Cada hilo tiene una prioridad que afecta a cómo el hilo es tratado por el sistema operativo.

**UUID.** Un Identificador universalmente único (universally unique identifier o UUID) es un identificador estándar usado en el desarrollo de software, estandarizado por la Open Software Foundation (OSF) como parte del entorno de computación distribuida (Distributed Computing Environment o DCE). La intención de los UUID es habilitar a los sistemas distribuidos un identificador de información único sin una importante coordinación central. Esto es, cualquiera puede crear un UUID y usarlo para identificar algo con una razonable confianza de que el identificador nunca será usado inintencionadamente por cualquiera para cualquier cosa.

**Handler.** Un Handler permite enviar y procesar mensajes y objetos ejecutables asociados a la cola de mensajes de un hilo de ejecución. Cada instancia de un Handler es asociada a un único hilo y a su cola de mensajes. Cuando se crea un nuevo Handler, se enlaza automáticamente con el hilo que lo creó y, a partir de ese momento, enviará mensajes y ejecutables a la cola de mensajes.





# 1. Introducción

---

## 1.1. Motivación

La aplicación desarrollada en el presente proyecto, AppTendance, está dirigida a las instituciones educativas que implementen la metodología estipulada por el plan Bolonia [1] y, por lo tanto, a cualquier institución en la Unión Europea, así como cualquier otro grupo que necesite apuntar la asistencia de sus componentes (entrenamientos deportivos, clases privadas...), y tiene como objetivo eliminar la necesidad de dedicar una parte del tiempo de la clase al control de la asistencia además de reducir los recursos materiales empleados para este control.

En el nuevo plan Bolonia se define el sistema de ECTS, un sistema que permite medir el trabajo que deben realizar los estudiantes para la adquisición de los conocimientos, capacidades, y destrezas necesarias para superar las diferentes materias de su plan de estudios. La actividad de estudio (entre 25 y 30 horas por crédito), incluye el tiempo dedicado a las horas lectivas, horas de estudio, tutorías, seminarios, trabajos, prácticas o proyectos, así como las exigidas para la preparación y realización de exámenes y evaluaciones, y es por este motivo por el que es necesario que los profesores lleven el control de las horas lectivas a las que asisten sus alumnos [2].

Hasta ahora, las instituciones que realizan un control de la asistencia de sus alumnos dejaban en manos del profesor la forma en la que realizar dicho control en sus clases, originando varios problemas como:

1. Las instituciones no dan unas reglas definidas, de forma que los profesores del mismo centro pueden usar la misma metodología, pero de formas distintas. Esto puede ocasionar equivocación en el alumno, o, en caso de que el profesor esté en desacuerdo con la metodología del centro, podría perder u omitir cierta información.
2. Para verificar las faltas de asistencia, el profesor debe ser capaz de identificar a cada estudiante, sin importar el número de clases de la que sea responsable ni el número de estudiantes en cada una de ellas.
3. No se establece un tiempo específico en el cual realizar el control de la asistencia, si no que se deja a discreción del profesor de cada asignatura. Además cada profesor no realiza el control a la misma hora cada día. Puede optar por cambiar el momento de la clase en el que pasar lista, ganando efectividad a la hora de controlar que los estudiantes abandonen el aula antes del final de la clase; pero que, sin embargo, puede ser confuso para los estudiantes.

Además, en la actualidad los teléfonos móviles han revolucionado la forma en la que nos comunicamos. Los Smartphones capaces de tomar fotos, grabar vídeos, crear documentos, reproducir música, así como realizar llamadas telefónicas, están disponibles a precios asequibles en todo el mundo. Incorporar las modernas tecnologías de comunicación en el entorno educativo permite al instructor no sólo impartir sus clases de forma más efectiva, sino también facilitar sus tareas administrativas, como se

ha podido comprobar en algunas instituciones que ya utilizan alguna de estas tecnologías para, por ejemplo, pasar lista en clase [3]. Apuntar la asistencia de los estudiantes presentes en una clase es una tediosa tarea que emplea mucho tiempo, especialmente si el número de estudiantes es muy grande. El instructor de la clase debe llamar a cada alumno por su nombre y el estudiante ha de confirmar su presencia alzando la mano; o bien el instructor hace circular una hoja entre los estudiantes confiando en que sean honestos y marquen únicamente su asistencia en la hoja, y no la de sus compañeros ausentes. Por todas estas razones, he decidido implementar en este proyecto un sistema de anotación de asistencia empleando los dispositivos móviles del profesor y de los alumnos que agilice este proceso, reduzca los recursos consumidos, y a la vez facilite la comunicación entre profesor y estudiante mediante la posibilidad de enviar preguntas al profesor de forma privada durante el transcurso de la lección.

## **1.1. Objetivos**

El objetivo de este proyecto es desarrollar una aplicación para Android que implemente un sistema de control de asistencia automático, agilizando el proceso de controlar la asistencia de los alumnos. Gracias a este proyecto, la carga y responsabilidad del profesor se verá reducida, ya que no es necesario que reconozca a todos los estudiantes desde el primer día de clase, ni buscar quién es el que falta, de forma que pueda concentrarse completamente en impartir la asignatura. Además, los datos de asistencia se podrán consultar y exportar fácilmente desde la base de datos del teléfono móvil del profesor, agilizando las consultas y evitando la pérdida de hojas de papel.

También se pretende mejorar la atención y el rendimiento de los estudiantes al incorporar la opción de enviar preguntas o mensajes al profesor durante el transcurso de la lección, de forma que el estudiante vea sus dudas resueltas en el momento, y sin temor a ser juzgado por los demás alumnos de la clase, ya que sólo el profesor sabrá la identidad del alumno que pregunta.

## **1.2. Estructura de la memoria**

El presente documento intenta resumir el trabajo de investigación y desarrollo llevado a cabo durante la realización de la aplicación Android de control automático de asistencia a clase. La estructura de la memoria se compone de los siguientes apartados:

### **1. Introducción.**

Este apartado trata la motivación del proyecto y los objetivos que pretende alcanzar.

### **2. Estado del arte.**

Incluye un estudio sobre proyectos similares que han sido desarrollados hasta el momento, y sobre las diferentes posibilidades y tecnologías consideradas a la

hora de desarrollar la aplicación. Además se redacta un resumen de las mejoras aportadas por el presente proyecto respecto a los trabajos existentes en la actualidad.

### **3. Diseño y desarrollo.**

En este capítulo se describen las tecnologías utilizadas para el desarrollo del proyecto: el protocolo de comunicación Bluetooth y el sistema operativo Android, así como el entorno de desarrollo en el que se ha diseñado y programado la aplicación: Eclipse, Android Development Tools y el simulador GenyMotion.

Se incluye además la descripción de la arquitectura interna de la aplicación y una descripción detallada de cada uno de los módulos que la componen, de la base de datos, y del funcionamiento de las conexiones Bluetooth.

### **4. Evaluación del sistema.**

En este apartado se explica el funcionamiento de la aplicación y se enumeran las pruebas realizadas durante el desarrollo de la misma, con sus resultados: pruebas sobre los módulos y actividades de la aplicación, sobre la base de datos, sobre las conexiones Bluetooth, sobre las distintas versiones y dispositivos en los que se puede instalar, y finalmente un informe sobre pruebas realizadas tras terminar el desarrollo de la aplicación simulando casos reales de uso.

### **5. Conclusiones y trabajo futuro.**

Finalmente este documento incluye un apartado de reflexión sobre el conjunto del trabajo realizado, así como algunas consideraciones o sugerencias para futuras actualizaciones de la aplicación desarrollada, o trabajos relacionados con esta.

### **6. Referencias.**

### **7. Anexos**

## 2. Estado del arte

---

En este apartado se repasan en primer lugar las técnicas usadas para el control de asistencia en otros sistemas existentes e implementados anteriormente, seguido de una descripción de las tecnologías disponibles para una aplicación como la que se quiere desarrollar en este proyecto.

### 2.1. Descripción de sistemas existentes

En la actualidad muchos sistemas y aplicaciones han sido desarrollados con el objetivo de automatizar el proceso de asistencia, pero casi ninguno ha conseguido satisfacer todas las necesidades. Hoy en día, la asistencia se apunta por lo general en un papel. Mediante el uso de los teléfonos móviles se consume menos tiempo y además ayuda a reducir el uso de papel, resultando ser una forma más ecológica de apuntar la asistencia. Existen varios pros y contras para cada una de las soluciones existentes, y creo que se pueden mejorar algunos aspectos, así como añadir más funciones que ayuden al profesor a impartir sus clases. A continuación enumero algunos ejemplos y las tecnologías que se han sugerido en ensayos, o usado en proyectos piloto, algunos de los cuales incluso se han puesto en funcionamiento en algunas universidades, separándolos según su funcionalidad: tecnologías empleadas a la hora de registrar la asistencia del estudiante y tecnologías empleadas para transmitir y almacenar los datos.

#### 2.1.1. Tecnologías para registrar la asistencia del estudiante

##### 2.1.1.1. Pasar lista de forma manual

Es el método empleado por la gran mayoría de aplicaciones disponibles en el mercado actualmente con ejemplos como Attendance [4], Mobile Attendance [5], Easy Attendance School College [6] y Attendance Manager [7], entre otras. Estas aplicaciones permiten cargar una lista de estudiantes desde una base de datos, y es el profesor quien debe llamar a los alumnos y marcar de forma manual su asistencia en la interfaz del teléfono móvil. Con esta metodología no se emplea papel y se centralizan todos los datos de asistencia, por lo que se centran en facilitar la organización de la información, ya que no es necesario traspasar los datos de las hojas de firmas. Sin embargo, el tiempo empleado en confirmar la asistencia es el mismo. Además, la única información que pueden recibir los alumnos es su informe de asistencia, siempre que cuenten con una conexión a internet y que la aplicación permita subir estos datos a un servidor remoto. Con este tipo de aplicaciones, no se ahorra nada de tiempo y el profesor sigue teniendo la misma responsabilidad de identificar qué alumno falta, por lo que no considero esta opción como apta para la implementación de este proyecto.

#### 2.1.1.2. Asociación de datos infrarrojos (IrDA)

Es una de las tecnologías más extendidas para la transmisión de información entre dispositivos inalámbricos [8]. Esta tecnología está basada en rayos luminosos que se mueven en el espectro infrarrojo. Los estándares IrDA soportan una amplia gama de dispositivos eléctricos, informáticos y de comunicaciones, permitiendo la comunicación bidireccional entre dos extremos a velocidades que oscilan entre los 9600 bit/s y los 4 Mbit/s. Esta tecnología que se encontraba en muchos ordenadores portátiles y teléfonos móviles de finales de los 90's y principios de la década del 2000, sobre todo en los de fabricantes líderes como Nokia y Ericsson, fue gradualmente desplazada por tecnologías como Wifi y Bluetooth [9]. Como se explica en el proyecto desarrollado por alumnos de la Universidad Pontificia de Salamanca [10], su funcionamiento se basa en un diodo situado a una distancia relativamente cercana, emitiendo un rayo de luz que alcanza el dispositivo. Se proveería a cada estudiante un emisor, por un coste de menos de 1€, que sería capaz de transmitir una señal portadora a un receptor situado en el aula y que al recibir dicha señal procedería a grabar la asistencia del correspondiente alumno. Sin embargo, el número de señales portadoras, la imposibilidad de mostrar al estudiante el resultado de la recepción de la señal, o la necesidad de establecer una línea de visibilidad entre el emisor y el receptor, son solo algunos de los problemas que dificultan el posible uso de esta tecnología para el proyecto.

#### 2.1.1.3. Identificación por radiofrecuencia (RFID)

En los proyectos en los que se hace uso de esta tecnología para registrar la asistencia de los estudiantes, como en los trabajos “*RFID Technology Based Attendance Management System*” [11] y “*Bluetooth Based Attendance Management System*” [12], publicados en el International Journal of Computer Science Issues y en el International Journal of Innovations in Engineering and Technology respectivamente, se debería expedir a cada estudiante una tarjeta RFID, y se colocaría un lector de tarjetas en el aula, el cual mandaría los datos a un servidor [12]. Esta es una tecnología viable, barata, simple de instalar y de mantener; sin embargo, se necesitaría proporcionar a los estudiantes algún tipo de información de “*feedback*” a través del dispositivo de cada aula, incrementando el tiempo necesario para confirmar la asistencia, de forma que no se resuelve el problema para el cual el actual proyecto había sido pensado. Además consideramos una gran desventaja el hecho de que, dado que para utilizar este sistema hay que proporcionar a cada alumno una tarjeta RFID así como instalar un lector de tarjetas en cada aula, existen una serie de costes iniciales que hay que afrontar antes de empezar a utilizar el sistema. Con la aplicación AppTendance desarrollada en este proyecto, el coste es cero, ya que parte de la base de que la mayoría de los alumnos poseen un dispositivo móvil Android.

#### 2.1.1.4. Comunicación de campo cercano (NFC)

Para resolver el problema del RFID, existe una tecnología que se está empezando a extender en Europa, Near Field Communication (NFC) [13]. Esta tecnología está presente en dispositivos cada vez más comunes entre los estudiantes, como son los Smartphones o las tabletas, y los operadores pueden proporcionar alternativamente tarjetas SIM (Subscriber Identity Module) con NFC incorporado, o bien artículos de bajo coste que lleven una tarjeta NFC (llaveros, pulseras, tarjetas... por

un coste inferior a 2€). NFC es un sistema de transmisión de datos que utiliza los principios de la tecnología RFID, es decir, una tecnología de corto alcance y alta frecuencia (13.56 MHz) que permite el intercambio de datos entre dispositivos [14]. La ventaja principal deriva de la distancia máxima permitida entre los dispositivos para realizar la conexión, que debe ser inferior a 10.16 cm (aunque la distancia máxima efectiva suele ser de aproximadamente 5 cm), de forma que se asegura la asistencia del alumno a la clase a la hora de realizar este intercambio de datos. Esta tecnología se ha implementado con éxito en un proyecto piloto de control de asistencia en la Universidad Pontificia de Salamanca [10]; sin embargo, considero que esta implementación tiene una serie de carencias funcionales que podrían ser útiles para su uso en las clases. Mediante el uso de NFC, no se puede establecer una conexión directa entre el profesor y sus alumnos más allá de la confirmación de su asistencia. Además, el sistema implementado en los proyectos piloto almacenan los datos en un servidor remoto, por lo que requiere una plataforma web de administración, así como un mantenimiento y dependencia del servidor y la conexión a internet. Por estos motivos, decidí no usar esta tecnología para la implementación de mi proyecto.

#### 2.1.1.5. Biometría



##### Reconocimiento ocular o dactilar

El reconocimiento de iris y de huella dactilar son dos de los métodos de identificación personal más seguros en Biometría. Para la implementación de estas tecnologías, como en los trabajos “*Wireless attendance management system based on iris recognition*” publicado en Scientific Research and Essays [15], “*An Automated Biometric Attendance Management System with Dual Authentication Mechanism Based on Bluetooth and NFC Technologies*” publicado en International Journal of Computer Science and Mobile Computing [16] o “*Wireless Fingerprint Based Student Attendance System*” del departamento de ingeniería electrónica del Instituto Nacional de Tecnología de Rourkela [17], se necesita instalar un lector dactilar o un scanner de iris junto al dispositivo encargado de realizar la comprobación. A la hora de confirmar la asistencia, el usuario debe acercar su dedo o su ojo al lector, y este, tras realizar el escaneo y las comprobaciones, anota su asistencia. Este método es muy costoso debido al precio de los dispositivos empleados para el reconocimiento. Además, cada alumno necesita un cierto tiempo de utilización del scanner, por lo que se produciría un cuello de botella y un coste de tiempo excesivo. Por estas razones, así como problemas con la privacidad de los estudiantes que no quieran registrar sus huellas dactilares o su iris, no considero adecuadas estas tecnologías para la realización del presente proyecto.



##### Reconocimiento facial

El reconocimiento facial es un método empleado desde hace bastante tiempo. A pesar de no ser tan preciso como otros métodos de identificación como el reconocimiento ocular o dactilar, el reconocimiento facial ha sido siempre objetivo de investigaciones debido a su naturaleza no invasiva y a que no requiere prácticamente ningún esfuerzo por parte del usuario. Mediante la

instalación de un circuito de vídeo, los estudiantes muestran su cara al entrar a clase, y con algoritmos de reconocimiento de imágenes, el sistema comprueba y anota su asistencia automáticamente. Estos sistemas, como el desarrollado en “*Study of Implementing Automated Attendance System Using Face Recognition Technique*” publicado en International Journal of Computer and Communication Engineering [18], reduce a cero el tiempo empleado en verificar la asistencia de los alumnos, y no crea ningún cuello de botella al usar un lector como en el caso del reconocimiento ocular o dactilar. Sin embargo, resulta costoso y trabajoso el hecho de instalar cámaras, y más caro aún implementar un sistema de reconocimiento que funcione de forma correcta, además de que no se cuenta con el *feedback* necesario para que el alumno se asegure de que su asistencia ha sido anotada correctamente.

#### 2.1.1.6 GPS-GPRS

En el sistema descrito en “*Mobile Phone Based Attendance*” publicado en Journal of Computer Engineering [19], el profesor anota la asistencia en el móvil y la manda a un servidor remoto donde las listas serían actualizadas automáticamente. Además, para evitar que el dispositivo del profesor registre la asistencia de sus alumnos por error, se incorpora un sistema de detección de la localización del dispositivo del profesor utilizando GPS. Gracias a este sistema, el profesor sólo será capaz de pasar lista cuando se encuentre en el aula, lo que no consideramos muy útil a la hora de evitar la falsificación del registro por parte de los alumnos.

Se contempló la posibilidad de usar la tecnología GPS para el proyecto descrito en esta memoria, modificándolo para que sean los dispositivos de los alumnos los que tengan que estar en el aula a la hora de confirmar su asistencia, pero la necesidad de contar con una conexión GPRS y la imposibilidad de establecer una comunicación bidireccional entre alumno y profesor descartaron dicha posibilidad.

#### 2.1.1.7 Bluetooth

El uso de la tecnología Bluetooth para el intercambio de datos entre dispositivos móviles es muy común hoy en día: intercambio de archivos como fotos o música, auriculares inalámbricos, etc. En trabajos como “*Bluetooth based attendance management*” [12], se combina esta tecnología con un lector de tarjetas RFID, de forma que el lector recoge la asistencia de los estudiantes, la envía a un ordenador, y éste se encarga de enviar los datos de asistencia a los dispositivos de los profesores. Sin embargo, el ejemplo “*Bluetooth Chat*” de la web de desarrollo de Android [20] y “*Bluetooth student registration system*” de Justin L. Brown [21], muestran que es posible el intercambio efectivo de información entre dispositivos móviles, creando una comunicación bidireccional. En el caso del trabajo desarrollado por Justin L. Brown, se crea un sistema de control de asistencia a clase utilizando únicamente el Bluetooth como método de verificación de la asistencia. Enviando la identificación personal del alumno al ordenador del profesor, se registra la presencia del alumno en el aula, permitiendo además intercambiar trabajos o proyectos, exámenes, etc.



De este modo, y sirviendo como inspiración los ejemplos antes mostrados, decidimos utilizar la tecnología Bluetooth para la implementación de este proyecto, adaptándolo de forma que pueda ser usado en dispositivos móviles.

## **2.1.2. Tecnologías para almacenar la información**

### **2.1.2.1. Plataforma web**

La mayoría de las soluciones que han sido planteadas al problema de automatizar el proceso de recogida de asistencia, como la propuesta en “*Mobile Based Attendance Management System*” [22], o en las aplicaciones de control de asistencia disponibles en Google Play Store (ver [4], [5], [6] y [7]) se basan en enviar los datos de asistencia a una base de datos remota, normalmente mediante el uso de internet (GPRS). En la actualidad, la mayor parte de las aplicaciones que se pueden encontrar en Google Play se basan en este principio: el profesor anota la asistencia en su dispositivo de forma manual, y los datos son enviados por internet a la base de datos remota que procesa y almacena las listas y estadísticas, y que puede ser consultada en cualquier momento mediante el ingreso a una plataforma web, tanto por profesores como por alumnos.

Este sistema centraliza los datos de asistencia de forma que no es necesario el uso de ningún soporte físico como hojas de papel, y mantiene la información actualizada y disponible tanto para el profesor como para los alumnos. Sin embargo, necesita de un servidor que albergue la base de datos, así como una conexión GPRS que permita transmitir estos datos desde el dispositivo del profesor. Es por estos inconvenientes por lo que optamos por no elegir esta tecnología para la implementación de este proyecto, ya que el proyecto pretende tener coste cero, y que pueda ser utilizado en cualquier dispositivo, sin necesidad de contratar una tarifa de datos.

### **2.1.2.2. Base de datos SQLite**

Con el objetivo de buscar elementos comunes a la mayoría de dispositivos móviles Android, decidimos evitar las conexiones a internet, evitando así servidores que mantener y la necesidad de contratar tarifa de datos. Por este motivo, decidimos que la mejor idea sería almacenar la información de las clases y alumnos de cada profesor en la base de datos interna de su propio dispositivo; así como la información personal relativa a cada estudiante en el suyo. De esta forma, aprovechando que todos los dispositivos Android incorporan la librería SQLite [23], podemos crear y utilizar una base de datos en la memoria interna del dispositivo mediante el lenguaje SQL de forma sencilla y utilizando muy pocos recursos del sistema, evitando a su vez posibles problemas en la seguridad de la información almacenada, ya que sólo se puede acceder a ésta a través de la aplicación instalada en el dispositivo del propietario, y no mediante una contraseña como en una aplicación web.



## **2.2. Aportaciones del sistema desarrollado**

El sistema planteado en este proyecto establece una comunicación bidireccional mediante Bluetooth entre los dispositivos móviles Android del profesor y sus alumnos, registrando la asistencia de éstos de manera automática, y almacenándola en la base de datos interna del dispositivo. Además, gracias a la bidireccionalidad de la comunicación, es posible enviar la confirmación del procedimiento al alumno, así como enviar y recibir mensajes entre alumno y profesor, sin necesidad de conexión a Internet y con coste cero.

Frente a las aplicaciones y sistemas implementados hasta el momento, la solución presentada en este proyecto aporta una serie de mejoras y nuevas posibilidades, tanto en el proyecto actual como en futuras mejoras y actualizaciones.

- **Ahorro de tiempo y recursos**

Para utilizar esta aplicación, sólo hace falta disponer de un dispositivo móvil con Android que soporte Bluetooth. En la actualidad, casi todos los dispositivos móviles soportan Bluetooth y, según un estudio de mercado realizado por ABI Research, en 2018 se espera que haya más de 2 billones de Smartphones [24]. Además, según los últimos datos ofrecidos por la consultora Kantar Worldpanel ComTech [25], a principios de 2014 Android es el sistema operativo con mayor cuota de mercado en el mundo, por lo que se supone que la gran mayoría de alumnos dispondrán de un dispositivo móvil con el que registrar su asistencia mediante este sistema.

Al automatizar la tarea de recoger la asistencia, tanto el profesor como los alumnos sólo tienen que preocuparse de iniciar su aplicación e introducir el código de confirmación que el profesor dé ese día. No hará falta esperar a que el profesor lea uno por uno todos los nombres de la lista, y no habrá riesgo de que se extravíe la hoja de firmas en la que los alumnos se apuntan. Además, la solución presentada es una alternativa ecológica, ya que evita el uso de hojas de papel.

- **No necesita conexión a Internet**

El sistema propuesto utiliza únicamente el protocolo de comunicación inalámbrico Bluetooth como medio para transmitir la información entre los dispositivos del profesor y los alumnos, por lo que no es necesario disponer de una tarifa de datos en ninguno de los dispositivos.

- **No necesita mantenimiento**

Los datos y las listas de asistencia se almacenan en el dispositivo del profesor, sin utilizar servidores remotos ni plataformas web que mantener. De esta forma, cualquier persona o institución puede utilizar la aplicación: una universidad, un colegio, o un profesor particular o monitor de algún tipo de

actividad al aire libre. Además, como los datos se almacenan en el propio dispositivo, no es necesario registro ni acreditación de la identidad del usuario al entrar en la aplicación. Cada usuario tiene acceso únicamente a los datos contenidos en el almacenamiento interno de su dispositivo móvil.

- **No necesita inversión inicial**

Frente a los sistemas que implementan tecnologías como NFC, IrDA, RFID o técnicas basadas en biometría, la solución propuesta en este proyecto no necesita de una inversión inicial para adquirir o instalar elementos como lectores de tarjetas, diodos receptores de señales infrarrojas o chips de radiofrecuencia. Basta con que los alumnos descarguen la aplicación y activen el Bluetooth de sus dispositivos móviles.

- **Permite el intercambio de información**

La aplicación desarrollada permite no sólo confirmar la presencia del alumno en la clase, sino cualquier intercambio de información con el dispositivo del profesor. De esta forma, con la misma aplicación se pueden enviar preguntas de forma privada al dispositivo del profesor durante el transcurso de la clase, o en futuras actualizaciones, realizar test de forma interactiva en cualquier momento.

- **Proporciona confirmación del procedimiento**

En la solución propuesta, el alumno recibe en su dispositivo móvil la confirmación de que su asistencia ha sido registrada correctamente, y automáticamente se guarda en la base de datos del dispositivo del alumno, de forma que éste puede consultar su asistencia a la asignatura en cualquier momento. En el caso de que se produzca algún error, el alumno lo ve en el momento de realizar el registro, y puede acudir al profesor para que éste solucione el problema si es responsabilidad suya, o para que anote la asistencia del alumno de forma manual si hay algún problema con los dispositivos.

- **Funcionamiento alternativo**

En el caso de que no todos los alumnos cuenten con un dispositivo Android en el aula, o que se produzca algún fallo en el proceso de registro, el profesor tiene la opción de pasar lista de forma manual, leyendo uno por uno los nombres de los alumnos y anotándolo mediante un tic en su dispositivo, de forma que la asistencia queda igualmente almacenada en su base de datos. Además, si algún alumno se retrasa, o bien presenta algún justificante en los días siguientes a su ausencia, el profesor tiene la opción de modificar sus listas de forma manual.

## 3. Diseño y desarrollo

Como solución al problema presentado, y habiendo estudiado todas las posibles tecnologías de las que disponemos (ver punto 2.1), decidimos diseñar una aplicación para dispositivos Android, estableciendo una comunicación inalámbrica entre los dispositivos del profesor y sus alumnos mediante Bluetooth, y almacenando la información y las listas de asistencia en la base de datos interna SQLite de cada dispositivo [Figura 1].

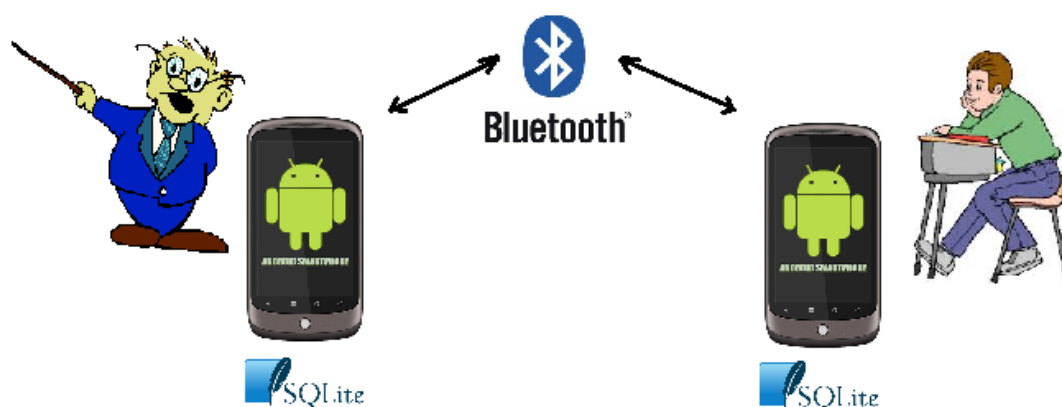


Figura 1: Esquema del funcionamiento básico de AppTendance.

### 3.1. Tecnologías utilizadas

#### 3.1.1. Comunicación inalámbrica: Bluetooth

El Bluetooth es una tecnología de comunicación inalámbrica bastante extendida basada en radio frecuencia que usa la banda de radio libre de ISM (Industrial, Scientific & Medical) a 2.4 GHz [26] y que pretende reemplazar los cables en la conexión de dispositivos, manteniendo altos niveles de seguridad. Las cualidades más importantes del Bluetooth son su ubicuidad, bajo consumo y bajo coste. Las conexiones entre dispositivos electrónicos con Bluetooth activo permiten la transmisión de información sin necesidad de cables dentro redes ad hoc de corto alcance conocidas como *piconets* [27]. Estos *piconets* se establecen de forma dinámica y automática cuando algún dispositivo Bluetooth entra o abandona el radio de proximidad, lo cual significa que se puede realizar la conexión fácilmente en cualquier momento. En cada *piconet* el dispositivo maestro puede comunicarse de forma simultánea con hasta siete dispositivos dentro de este *piconet*, y cada dispositivo puede también pertenecer a varios piconets de forma simultánea. Esto significa que las formas de conectar los dispositivos Bluetooth son casi ilimitadas.

El salto de frecuencia adaptable (AFH) que incorpora la tecnología Bluetooth fue diseñada para reducir la interferencia entre las tecnologías inalámbricas que

comparten el espectro de 2.4 GHz, mediante la detección de otros dispositivos dentro del espectro y evitando las frecuencias que éstos utilizan. Este salto adaptable entre 79 frecuencias a intervalos de 1 MHz proporciona un alto grado de inmunidad frente a las interferencias y además permite una transmisión más eficiente dentro del espectro, incluso si se usan otros métodos de transmisión de datos a la vez que el Bluetooth.

Sin embargo, el intercambio de datos no requiere que el emisor y el receptor estén en la misma sala, ya que se puede establecer una comunicación Bluetooth desde 10 hasta 30 metros de distancia del receptor, por lo que no es posible demostrar la presencia en las cercanías de un punto físico determinado.

No obstante, el uso del Bluetooth nos ofrece una posibilidad que no ofrece ninguna de las otras tecnologías estudiadas: la transmisión de datos de forma bidireccional entre emisor y receptor. De esta forma, el alumno recibe la confirmación de que su asistencia ha sido registrada de forma inmediata en su dispositivo. Además, no se necesita una conexión a internet, de forma que no es necesario contar con una tarifa de datos para tener acceso al servicio, ni un servidor remoto que almacene los datos el cual necesite mantenimiento o algún tipo de plataforma de administración.

Esta posibilidad de comunicación bidireccional puede ser aprovechada, y llevar más lejos el sistema automático de asistencia, incorporando la opción de responder sencillos test en clase, o enviando un mensaje anónimo al profesor con alguna pregunta, sin necesidad de que el resto de alumnos se enteren y sin miedo a ser juzgado por ello, motivo por el cual muchos alumnos no realizan preguntas durante la clase.

Viendo el abanico de posibilidades que ofrece la tecnología Bluetooth, y viendo que otros proyectos se han servido de esta tecnología de forma exitosa (ver referencias [21] y [12]), decidimos que sería la elegida para implementar el proyecto de recogida automática de asistencia a clase, ya que el único inconveniente (imposibilidad de demostrar que el dispositivo del alumno se encuentra dentro del aula) puede ser solventado fácilmente mediante la introducción de un código de control que el profesor decidirá y comunicará a los estudiantes dentro del aula a la hora de pasar lista.

### **3.1.2. Sistema operativo: Android**

La aplicación propuesta en este proyecto está programada para dispositivos con sistema operativo Android. Android es un sistema operativo basado en el kernel de Linux diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes o tabletas. Android, al contrario que otros sistemas operativos para dispositivos móviles como iOS o Windows Phone, se desarrolla de forma abierta y se puede acceder tanto al código fuente como a la lista de incidencias donde se pueden ver problemas aún no resueltos y reportar problemas nuevos. Las aplicaciones se desarrollan habitualmente en el lenguaje Java con Android Software Development Kit (Android SDK), el cual es provisto por Google de forma gratuita [28]. Además, las aplicaciones desarrolladas para Android pueden ser publicadas gratuitamente por los desarrolladores de forma que cualquier usuario de Android puede descargarla desde “Play Store”. De esta manera, las aplicaciones tienen una fácil difusión entre los usuarios de Android de cualquier parte del mundo. A principios de octubre de 2010, Google agregó 20 países a su lista de lugares geográficos donde los desarrolladores

pueden enviar aplicaciones. Para mediados de octubre, la compra de aplicaciones estaba disponible en un total de 32 países [29].

En cuanto a la cuota de mercado, Android supera por mucho a iOS y Windows Phone. En noviembre de 2013 Andy Rubin (ex CEO de Android y vicepresidente de ingeniería en Google) dijo que se activaban 1.500.000 dispositivos diariamente frente a los 700.000 dispositivos diarios de Diciembre de 2011 [30]. En abril de 2013 se hizo público que Android alcanzó el 92% en ventas de nuevos *Smartphones* para el trimestre comprendido entre diciembre 2012 y febrero 2013 en España, seguido de iOS con un 4.4% [31]. Según el estudio de mercado realizado el 28 de abril de 2014 por Kantar Worldpanel, Android lidera con un 88,6% las ventas en España [Figura 2].

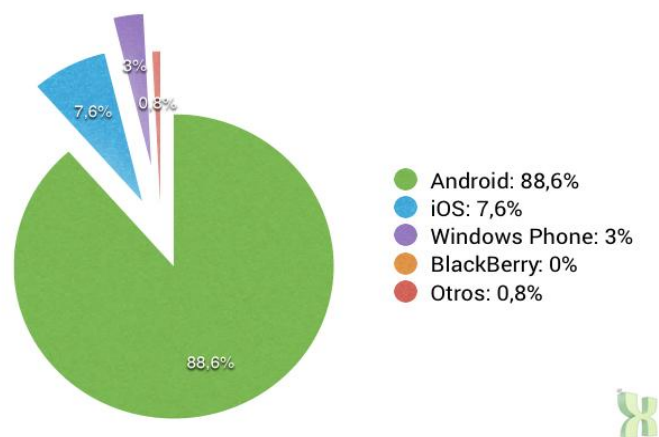


Figura 2: Cuota de mercado de Sistemas Operativos en España. Kantar Worldpanel.

En cuanto a la cuota de mercado a nivel mundial, Android también es líder. España es el país con mayor cuota de mercado de dispositivos Android, seguido de Alemania y China. Pese a una ligera disminución de esta cuota de mercado a favor de iOS y Windows Phone, Android sigue siendo el sistema operativo para Smartphones más vendido [Tabla 1].

### Smartphone OS Sales Share (%)

Germany	3 m/e Apr 2013	3 m/e Apr 2014	% pt. Change	USA	3 m/e Apr 2013	3 m/e Apr 2014	% pt. Change
Android	74.9	78.5	3.6	Android	51.8	59.1	7.3
BlackBerry	0.4	0.2	-0.2	BlackBerry	0.7	1.0	0.3
iOS	15.7	14.1	-1.6	iOS	41.4	34.6	-6.8
Windows	6.6	6.9	0.3	Windows	5.6	4.7	-0.9
Other	2.4	0.3	-2.1	Other	0.6	0.6	0.0
GB	3 m/e Apr 2013	3 m/e Apr 2014	% pt. Change	China	3 m/e Apr 2013	3 m/e Apr 2014	% pt. Change
Android	57.2	58.2	1.0	Android	71.9	79.8	7.9
BlackBerry	5.5	1.6	-3.9	BlackBerry	0.3	0.0	-0.3
iOS	28.0	30.2	2.2	iOS	23.7	17.5	-6.2
Windows	8.6	9.5	0.9	Windows	2.2	0.8	-1.4
Other	0.6	0.5	-0.1	Other	1.9	1.9	0.0
France	3 m/e Apr 2013	3 m/e Apr 2014	% pt. Change	Australia	3 m/e Apr 2013	3 m/e Apr 2014	% pt. Change
Android	68.0	66.6	-1.4	Android	64.1	60.8	-3.3
BlackBerry	3.8	1.1	-2.7	BlackBerry	0.6	0.9	0.3
iOS	16.9	20.2	3.3	iOS	28.1	30.1	2.0
Windows	6.2	10.1	3.9	Windows	5.0	6.7	1.7
Other	5.1	2.0	-3.1	Other	2.2	1.5	-0.7
Italy	3 m/e Apr 2013	3 m/e Apr 2014	% pt. Change	Japan	3 m/e Apr 2013	3 m/e Apr 2014	% pt. Change
Android	66.7	72.5	5.8	Android	44.4	43.6	-0.8
BlackBerry	2.0	1.0	-1.0	BlackBerry	0.8	0.0	-0.8
iOS	16.5	13.4	-3.1	iOS	49.8	55.4	5.6
Windows	10.5	11.8	1.3	Windows	0.6	1.0	0.4
Other	4.3	1.2	-3.1	Other	4.4	0.0	-4.4
Spain	3 m/e Apr 2013	3 m/e Apr 2014	% pt. Change	EU5	3 m/e Apr 2013	3 m/e Apr 2014	% pt. Change
Android	92.0	86.1	-5.9	Android	70.8	72.4	1.7
BlackBerry	0.1	0.0	-0.1	BlackBerry	2.4	0.8	-1.7
iOS	4.8	8.7	3.9	iOS	17.2	17.5	0.2
Windows	1.7	4.6	2.9	Windows	6.9	8.4	1.6
Other	1.4	0.6	-0.8	Other	2.7	0.9	-1.8

Tabla 1: Cuota de mercado de los distintos sistemas operativos para Smartphone a nivel mundial. Kantar Worldpanel, Abril 2014.

Hasta ahora, Android cuenta con 19 niveles de API, siendo el último de ellos la versión 4.4 (Kit Kat), distribuida desde el 13 de Octubre de 2014. Esta fragmentación entre los distintos niveles de API es un problema para los desarrolladores, ya que distintos niveles de API tienen distintas características y compatibilidades [Tabla 2] [Figura 3].

Versión	Código	API	Cuota de mercado
2.2	Froyo	8	0.8%
2.3.3 - 2.3.7	Gingerbread	10	14.9%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	12.3%
4.1.x	Jelly Bean	16	29.0%
4.2.x		17	19.1%
4.3		18	10.3%
4.4	KitKat	19	13.6%

Tabla 2: Cuota de mercado de las distintas versiones y niveles de API de Android.  
Android Developers.

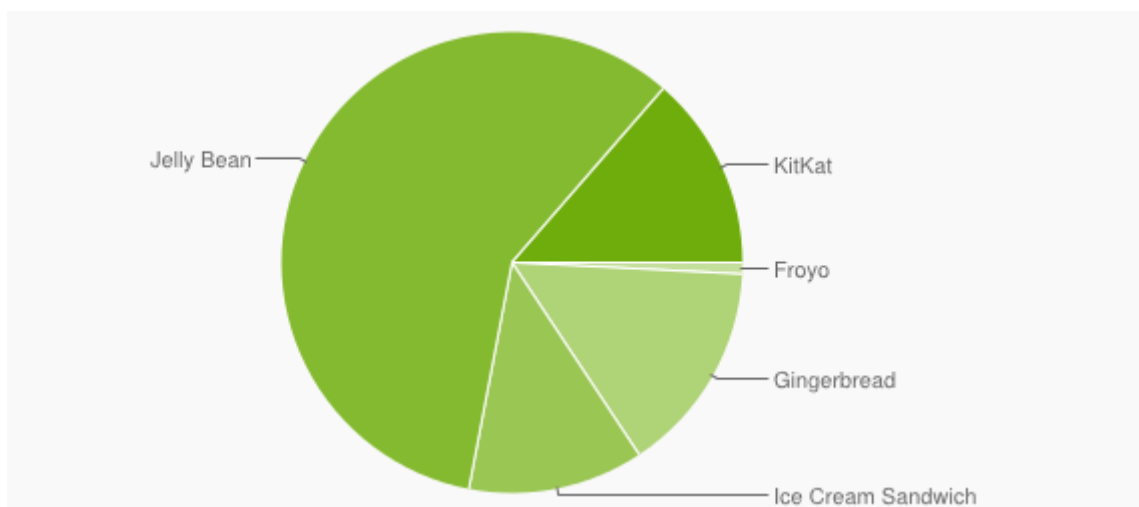


Figura 3: Distribución de la cuota de mercado de Android según versiones.  
Android Developers.

Los datos mostrados en las figuras fueron tomados el 4 de Junio de 2014. Las versiones con una distribución inferior al 0.1% no se incluyen en las gráficas. Además, los datos fueron recogidos a través de la nueva aplicación Google Play Store App, que soporta sólo versiones posteriores a la 2.2., por lo que los dispositivos con versiones anteriores no son incluidos. En Agosto de 2013, los dispositivos con versiones anteriores a la 2.2 suponían un 1% de todos los dispositivos registrados en los servidores de Google.



En el caso del proyecto desarrollado, AppTendance, soporta APIs a partir del nivel 14. De esta forma, el problema de la fragmentación de versiones de Android sólo excluiría a menos del 16% de los dispositivos que utilizan este sistema operativo; y dado que estos dispositivos cuentan con las versiones más antiguas, tenderán a reducir este porcentaje de cuota de mercado con el tiempo, siendo reemplazados por las nuevas versiones.

### 3.2. Entorno de desarrollo

En la actualidad existen varios IDEs para desarrollar aplicaciones Android, normalmente basados en *Eclipse* o en *NetBeans*. Un IDE (del inglés *integrated development environment*), o entorno de desarrollo integrado, es un programa informático compuesto por un editor de código, un compilador, un depurador y un constructor de interfaz gráfica [32]. En la actualidad, los IDEs más extendidos para el desarrollo de aplicaciones Android son Eclipse y Android Studio [33], este último desarrollado por Google y basado en el software para Java de JetBrains IntelliJ [36]. He elegido Eclipse como entorno de desarrollo, ya que cuenta con numerosas herramientas para la depuración, así como para el diseño de la interfaz gráfica, y además tengo experiencia en su manejo gracias a trabajos anteriores.

Para desarrollar aplicaciones Android en Eclipse es necesario incluir en el entorno de desarrollo un *plugin* llamado *Android Development Tools* (ADT) [28], que permite, entre otras cosas, crear y configurar nuevos proyectos, diseñar gráficamente la interfaz de usuario de la aplicación y exportar los paquetes de aplicaciones (APK) para su distribución. Además, dentro del entorno de desarrollo de Eclipse se incluye un emulador que permite realizar pruebas simulando las características de distintos dispositivos y versiones de Android como si se tratara de un dispositivo físico real. Sin embargo, las pruebas de la aplicación se han realizado en parte con otro emulador de dispositivos Android: GenyMotion [35], cuya aceleración OpenGL consigue un mayor rendimiento y rapidez en las simulaciones, además de permitir iniciar varios dispositivos virtuales a la vez, activar la opción de visualizar la simulación a pantalla completa y es capaz de controlar los sensores del dispositivo, como el GPS, la rotación del dispositivo, la cámara o el nivel de batería [Figura 4].





Figura 4: simulador GenyMotion emulando un Samsung Galaxy S2.

### 3.3. Arquitectura del sistema

El sistema de control de asistencia automático AppTendance utiliza el adaptador Bluetooth de los dispositivos móviles Android para establecer una comunicación entre los dispositivos del profesor y sus alumnos, almacenando la información en la base de datos interna SQLite de cada dispositivo, además de proporcionar al profesor la posibilidad de importar o exportar estos datos a archivos en formato CSV en la memoria externa del dispositivo.

La arquitectura definida para el sistema de control automático de asistencia se puede dividir en cinco módulos según su funcionalidad y su interacción con el entorno [Figura 5]: actividades del profesor, actividades del alumno, módulo de conexión Bluetooth, adaptador de la base de datos y el módulo de lectura y escritura de ficheros con formato CSV.

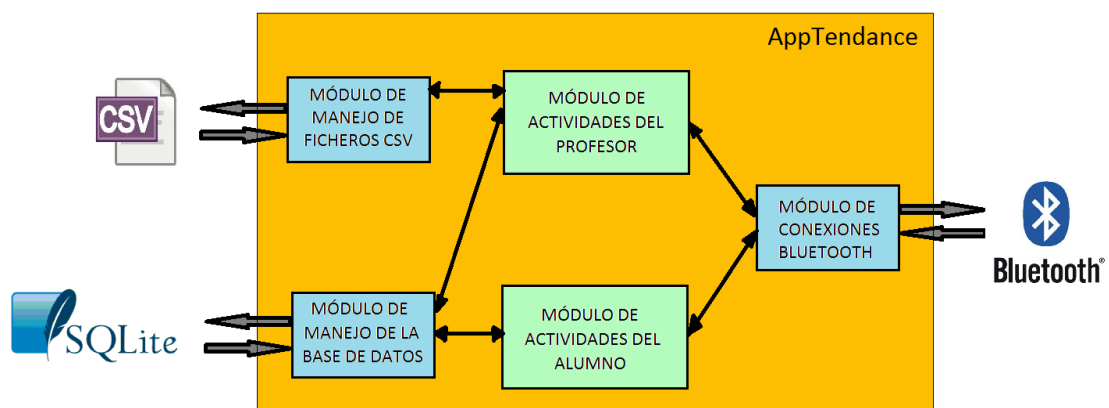


Figura 5: Esquema de la organización de los módulos de AppTendance.

### 3.3.1. Módulo de actividades del profesor

Este módulo agrupa todas las actividades y clases necesarias para que el usuario entre en la aplicación como profesor. Cada actividad representa una pantalla, y cada una de estas actividades tiene una función y un *layout* asociado. Estas actividades permiten al usuario navegar y elegir entre las distintas funcionalidades de la aplicación [Figura 6].

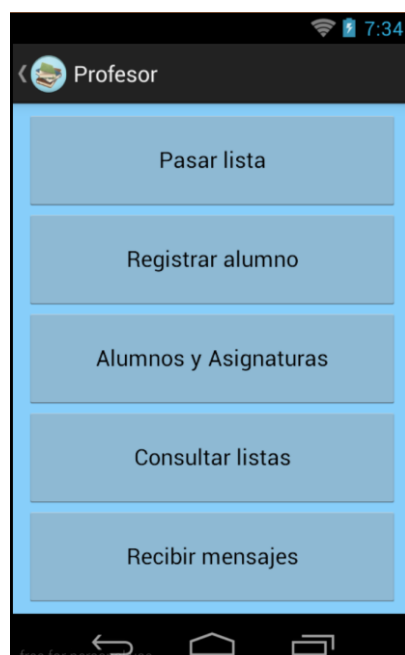


Figura 6: Pantalla de opciones del profesor.

#### ➤ Editar, añadir y borrar asignaturas.

El usuario (profesor) puede crear y manejar sus asignaturas: nombre, descripción y alumnos matriculados [Figura 7]. Además, se ofrecen dos formas de registrar alumnos en las asignaturas: bien de forma manual introduciendo directamente en el dispositivo los nombres, apellidos y DNI de cada uno de los alumnos, o bien importando los datos de un fichero en formato CSV copiado en la memoria externa que contenga esta información, utilizando para ello el módulo de manejo de ficheros CSV. Toda esta información relativa a alumnos y asignaturas queda almacenada en la base de datos interna del dispositivo, de forma que puede ser consultada o editada en cualquier momento.

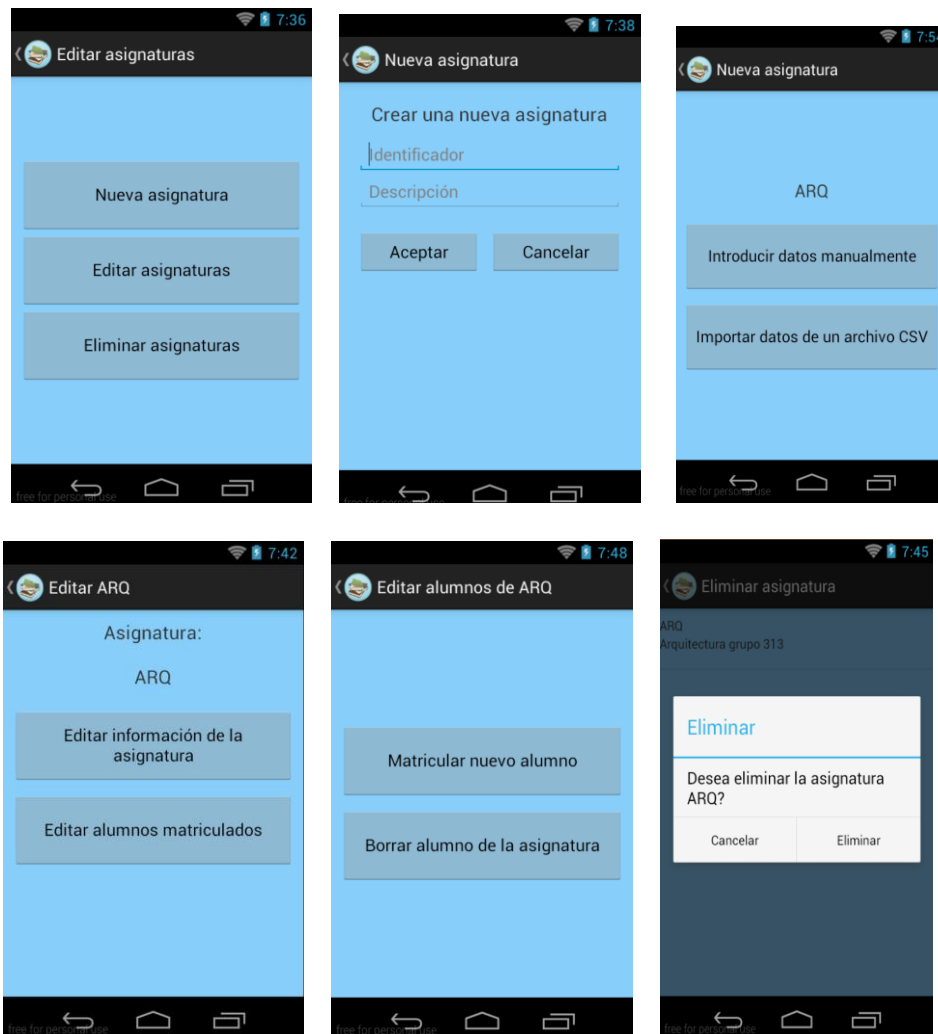


Figura 7: Distintas pantallas de opciones para crear, editar o eliminar asignaturas del dispositivo del profesor.

### ➤ **Editar, añadir y borrar alumnos.**

De forma similar al punto anterior, el profesor puede editar los datos de alguno de sus estudiantes, o bien cambiar las asignaturas de las que está matriculado, o agregar manualmente un nuevo alumno y registrarle en alguna de las asignaturas creadas con anterioridad [Figura 8].

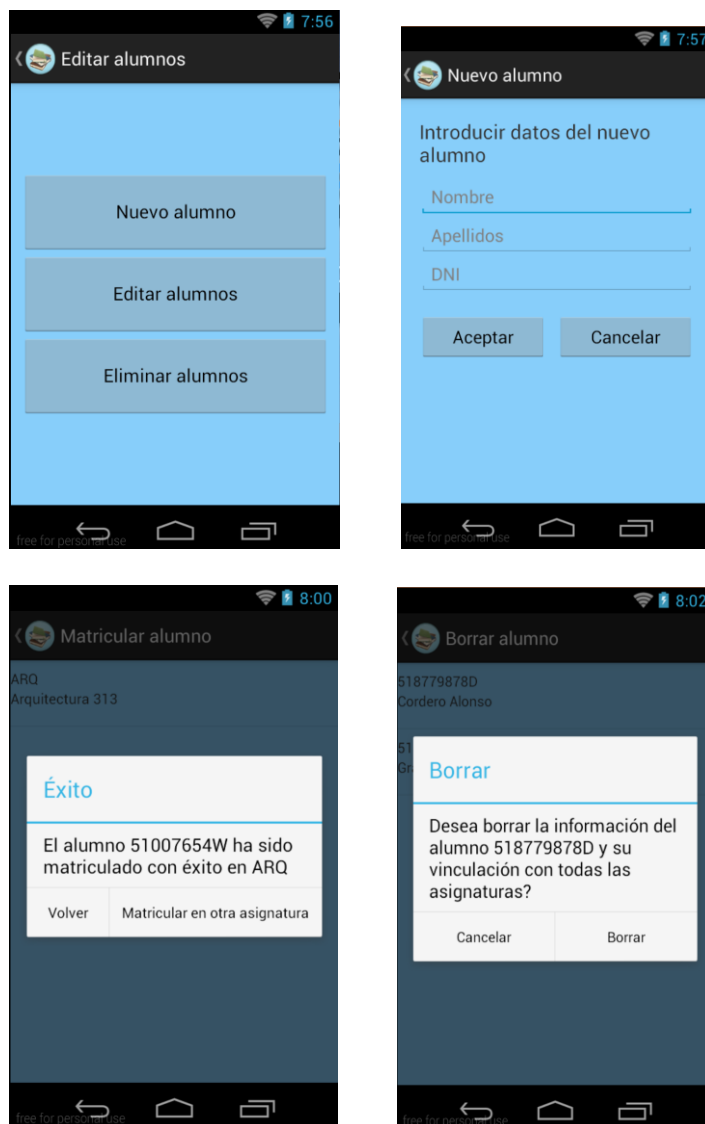


Figura 8: Distintas pantallas de opciones para crear, editar o eliminar alumnos del dispositivo del profesor.

### ➤ Registrar MAC de alumno

El funcionamiento del sistema de asistencia desarrollado se basa en el registro de la identidad de los dispositivos que utilizan los alumnos [Figura 9]. Esta identidad se garantiza mediante la dirección MAC única de cada dispositivo. En las actividades de este módulo, el profesor configura el adaptador Bluetooth de su dispositivo para escuchar conexiones entrantes mediante el módulo que maneja las conexiones Bluetooth. Cuando un alumno desea registrarse en la asignatura, debe enviar en este momento su dirección MAC y, tras comprobar en la base de datos que el alumno efectivamente está matriculado en la asignatura, y su identificación (DNI) coincide con la identificación introducida por el profesor, esta dirección MAC queda almacenada junto a la información del alumno en la base de datos, de forma que en el futuro no podrá confirmar la asistencia de ningún otro alumno desde su dispositivo, evitando así posibles fraudes. En el caso

de que el alumno cambie de dispositivo durante el curso, lo único que tendrá que hacer es instalar la aplicación en el nuevo dispositivo y realizar el registro de su nueva dirección MAC en el dispositivo del profesor, sin afectar a los registros de asistencia anteriores.

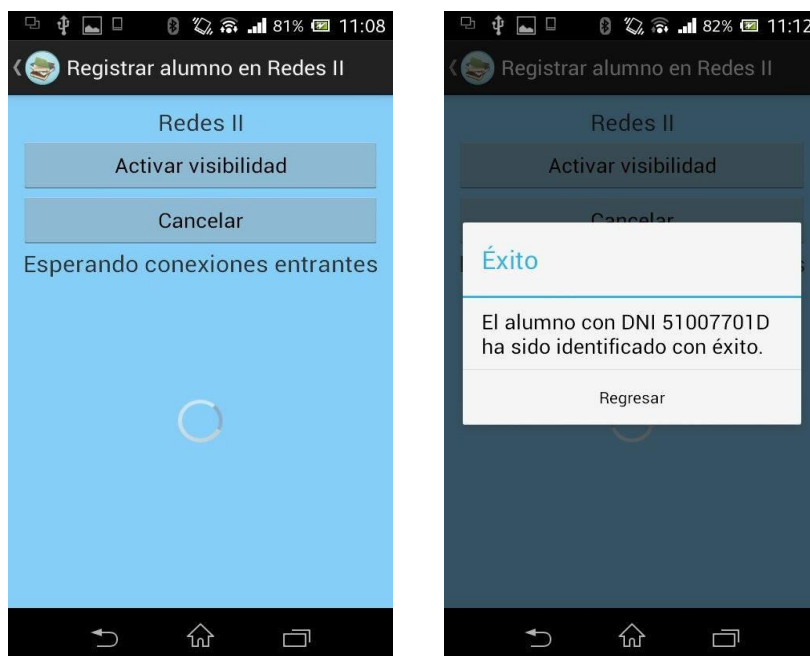


Figura 9: Pantalla de registro de alumnos y mensaje de confirmación del registro.

### ➤ Pasar lista

Este módulo es el que concentra las actividades que desarrollan la finalidad principal de la aplicación. Se ofrecen dos posibilidades a la hora de pasar lista: conectar mediante Bluetooth con los dispositivos de los alumnos, o anotarla de forma manual [Figura 10].

En el caso de que el profesor elija pasar lista de forma manual, se le presentará por pantalla un listado con la información de los alumnos en orden alfabético y un cuadrado al lado de cada nombre, de forma que el profesor solo tiene que anotar los alumnos presentes mediante un tic en la fila de su nombre. Al acabar, se mostrará una lista con el resumen y la estadística de la asistencia de ese día, y también se dará la opción de modificar esta lista.

A la hora de pasar lista utilizando conexiones Bluetooth, el profesor debe elegir la asignatura e introducir un código de control para evitar fraudes de alumnos que intenten confirmar su asistencia desde fuera de la clase aprovechando el alcance de las comunicaciones por Bluetooth. Después de comunicar el código de control a sus alumnos, utiliza el módulo de

conexiones Bluetooth de la aplicación para configurar el adaptador del dispositivo a la espera de conexiones entrantes. Según vaya recibiendo conexiones de los alumnos, comprobará que la dirección MAC de la comunicación coincida con la dirección registrada para el alumno en la base de datos interna del dispositivo. Si es correcta, y el alumno efectivamente está registrado en la asignatura, y el código de control enviado por el alumno coincide con el código facilitado por el profesor, la asistencia queda almacenada en la base de datos y se muestra por pantalla en tiempo real una lista con los alumnos que van confirmando su asistencia. Una vez completado este proceso, se envía un mensaje de confirmación (o del error producido) al dispositivo del estudiante, y se cierra la conexión para aceptar más conexiones de los demás alumnos.

Cuando el profesor decida terminar de pasar lista, el dispositivo deja de escuchar conexiones entrantes y muestra por pantalla una lista de la asistencia recogida, completando la tabla con ausencias para los alumnos registrados en la asignatura que no han confirmado su asistencia por Bluetooth. En este punto, se ofrece la opción de modificar la lista de forma manual para los casos en los que algún alumno no disponga de la aplicación o de un dispositivo Android, o para aquéllos que lleguen con retraso a clase. Además, el profesor puede pasar lista varias veces en el mismo día para la misma asignatura, de forma que sólo las ausencias se verán reemplazadas cuando el alumno confirme asistencia en una segunda ronda. Así, si varios alumnos llegan tarde a clase, se puede utilizar la ventaja del uso de Bluetooth para ahorrar tiempo.

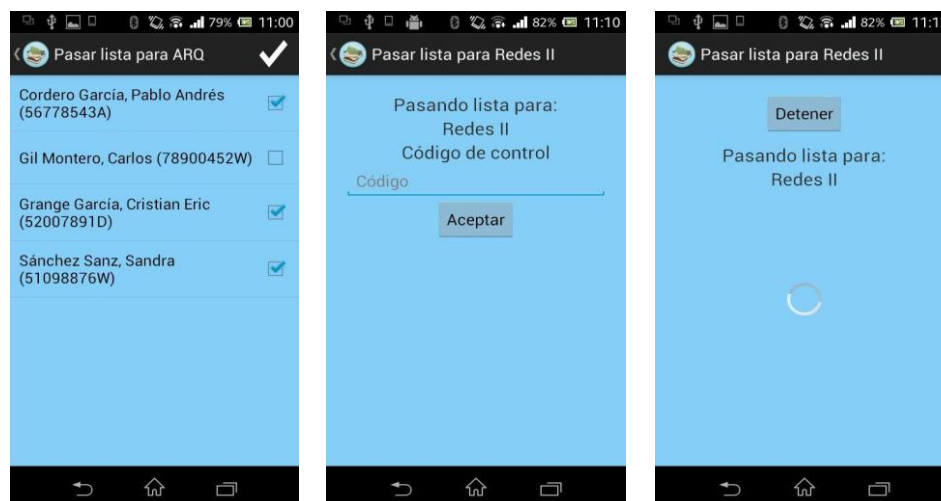


Figura 10: Distintas pantallas para pasar lista: de forma manual y mediante Bluetooth.

### ➤ Consultar listas de asistencia

El profesor tiene también la opción de consultar las listas de asistencia de todas sus asignaturas [Figura 11]. La aplicación ofrece la opción de filtrar las listas de asistencia por asignatura, por fecha o por estudiante [Figura 12]. En el caso de que el profesor quiera ver la asistencia total de la asignatura, se mostrará en una tabla toda la asistencia recogida hasta el momento para esta asignatura, agrupada según las fechas en las que se recogió esta asistencia originalmente, junto con la estadística global de asistencia a esta asignatura. Además, el profesor tiene la posibilidad de exportar estas listas a la memoria externa del teléfono en un fichero en formato CSV, de forma que pueda sacar esta información del dispositivo y copiarla a un ordenador, o una memoria externa, y abrir y consultar estos ficheros con programas de edición de hojas de cálculo tales como Microsoft Excel.

En el caso de que quiera consultar la asistencia de un día en concreto, sólo ha de seleccionar la fecha exacta que quiere consultar, y se mostrará el listado de los alumnos y su asistencia en ese día concreto.

Si lo que quiere es consultar la asistencia de un alumno en concreto a una de sus asignaturas, se mostrará la estadística de faltas y asistencias de dicho alumno, así como un listado completo de las fechas en las que se recogió asistencia y si el alumno estuvo presente o no.

De este modo, mediante una sencilla consulta a la base de datos, el profesor puede saber rápidamente el porcentaje de asistencia de sus alumnos, o si debe advertir a alguno de ellos de que ha llegado a su límite de ausencias. Además, se ofrece la posibilidad de modificar estas listas en el caso de que algún alumno presente un justificante de alguna ausencia en el pasado.

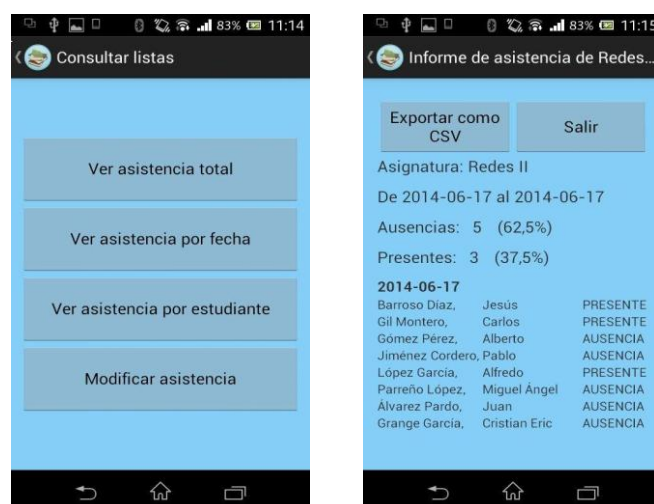


Figura 11: Pantalla de opciones para mostrar las listas de asistencia y lista de asistencia de una asignatura.

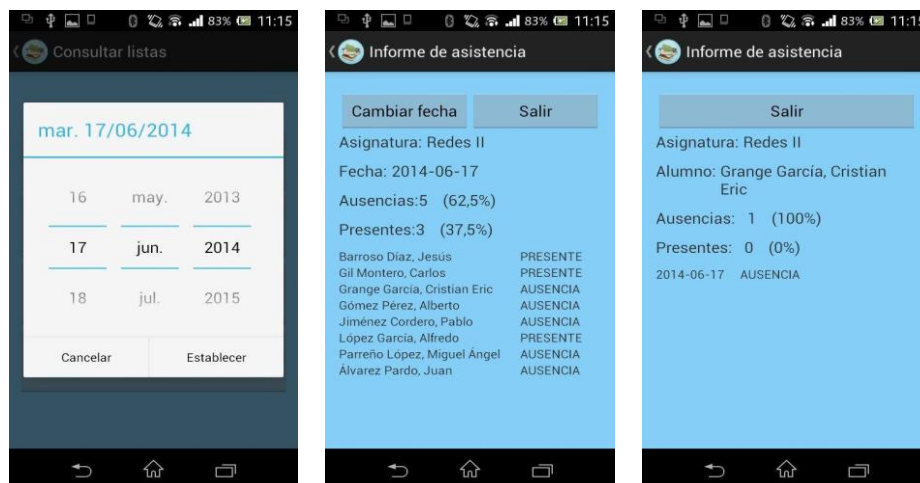


Figura 12: Pantalla de selección de fecha para consultar listas de asistencia, y resultados de búsqueda en lista de asistencia por fecha y por alumno.

### ➤ Recibir preguntas de los estudiantes

Este módulo permite al profesor seleccionar la asignatura que está impartiendo en ese momento, y se encarga de configurar el adaptador de Bluetooth para escuchar conexiones entrantes de los alumnos registrados en dicha asignatura [Figura 13]. Cuando algún alumno envíe una pregunta o mensaje, éste será mostrando por pantalla junto con una alerta sonora, de modo que el profesor que imparte la clase lo escuche y atienda el mensaje de forma inmediata, respondiendo la duda a la clase, o repitiendo los detalles de la lección que no han quedado claros.

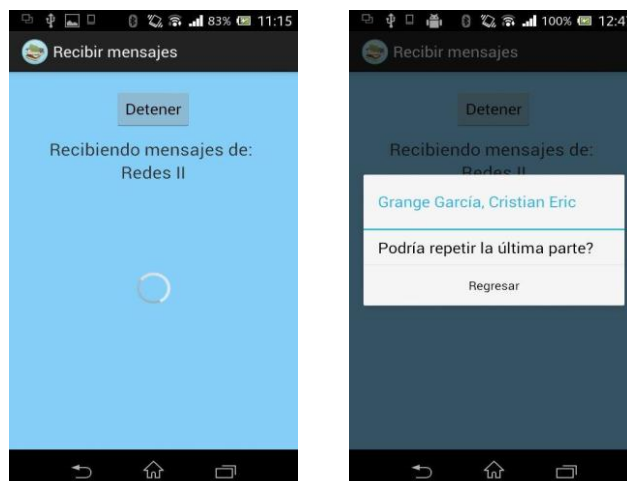


Figura 13: Pantalla de recepción de mensajes del profesor y alerta de nuevo mensaje.



### 3.3.2. Módulo de actividades del alumno

Al igual que el módulo anterior, este módulo agrupa las actividades por las cuales navega el usuario en el caso de utilizar la aplicación como un alumno [Figura 14].

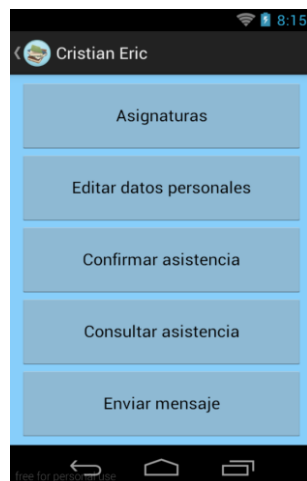


Figura 14: Pantalla de opciones del alumno.

#### ➤ Información personal

Al utilizar la aplicación por primera vez, el alumno debe introducir su información personal: nombre, apellidos y DNI [Figura 15]. Esta información queda almacenada en las Preferencias del dispositivo. A partir de este momento, la barra de actividad queda personalizada con el nombre del estudiante, y siempre que intente registrar su dispositivo para registrarse en una nueva asignatura, serán estas credenciales las que se usen. Sin embargo, el alumno puede cambiar esta información en cualquier momento desde el menú principal de la actividad, pero debe avisar a sus profesores para que modifiquen sus datos y vuelvan a registrar su dispositivo.



Figura 15: Pantalla de edición de datos personales del alumno.

### ➤ Agregar, editar o eliminar asignaturas

El alumno puede agregar nuevas asignaturas [Figura 16] o modificar sus asignaturas registradas siempre que quiera: puede editar el nombre de la asignatura, así como eliminar alguna asignatura de su base de datos [Figura 17]. Hay que tener en cuenta que el hecho de eliminar una asignatura del dispositivo del alumno no implica que se desmatricule de dicha asignatura, por lo que deberá hablar con el profesor para que elimine su información de su dispositivo y desmatricularle formalmente.

En el caso de querer registrarse en una asignatura nueva, debe para ello emparejar su dispositivo con el dispositivo del profesor que imparta la asignatura mediante una petición de enlace por Bluetooth. De este modo, una vez que ambos aceptan el vínculo, el nombre y la dirección del dispositivo emparejado queda almacenada en la memoria, y no hará falta pedir confirmación en futuras conexiones Bluetooth. A continuación, el alumno debe registrar su dispositivo. Para ello, profesor y alumno deben acordar utilizar la aplicación al mismo tiempo, de modo que el dispositivo del alumno envía su DNI y su dirección MAC al dispositivo del profesor, que debe usar la actividad de registrar alumno, con su adaptador de Bluetooth configurado como escuchador de conexiones entrantes. Si su identificación es correcta, y aparece en la base de datos del dispositivo del profesor como alumno de esta asignatura, se envía un mensaje de confirmación al alumno, que puede proceder a guardar la asignatura en la base de datos de su dispositivo, junto a la dirección MAC del profesor de la asignatura, de forma que en el futuro no tiene que volver a realizar este emparejamiento y registro, sino simplemente elegir el dispositivo con el que quiere conectar para confirmar su asistencia.

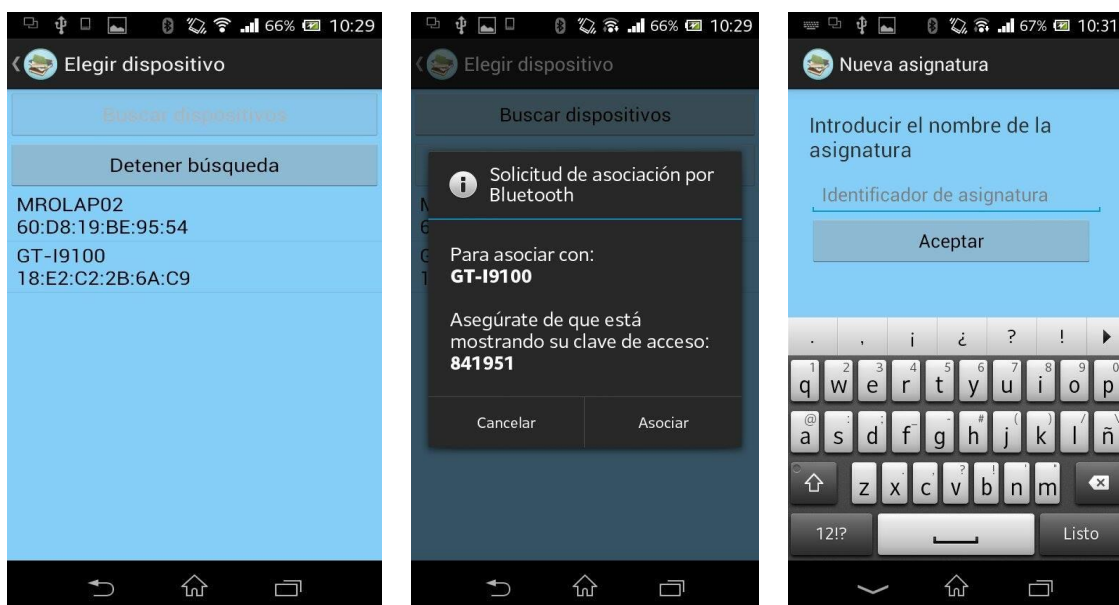


Figura 16: Proceso de búsqueda y emparejamiento del dispositivo del profesor de la asignatura y adición de ésta a la base de datos.

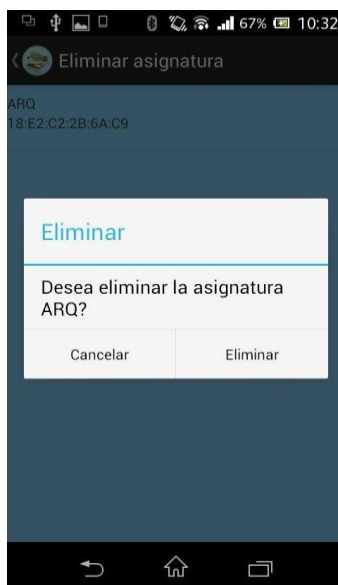


Figura 17: Confirmación para borrar una asignatura del dispositivo del alumno.

#### ➤ Confirmar asistencia

Gracias al sistema desarrollado en este proyecto, el proceso de confirmación de la asistencia es rápido y sencillo [Figura 18]: sólo tiene que seleccionar la asignatura de entre las asignaturas guardadas en la base de datos e introducir el código de control proporcionado por el profesor. A continuación el dispositivo intentará conectar con el dispositivo del profesor hasta que éste último esté libre y acepte la conexión, sin que el alumno deba interactuar en ningún momento con el dispositivo. Una vez establecida la conexión, el alumno recibirá un mensaje con el estado de su petición, confirmando que su asistencia ha sido registrada, o informándole del error cometido (si ha introducido un código de control incorrecto, si no aparece en la lista de alumnos registrados, si el alumno intenta conectarse desde un dispositivo que no es el suyo, si su asistencia ya ha sido registrada anteriormente o si ha habido algún fallo durante la conexión). Cuando el alumno recibe una confirmación positiva, la asistencia queda almacenada en la base de datos del alumno para su posterior consulta.

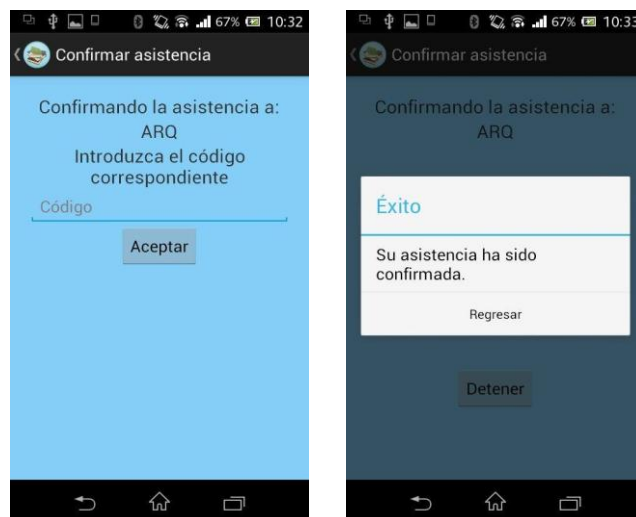


Figura 18: Pantallas de confirmación de la asistencia del estudiante.

### ➤ Consultar asistencia propia

El alumno puede también consultar su asistencia a una asignatura [Figura 19]. Para ello simplemente debe elegir la asignatura que desea consultar, y se mostrará en pantalla el número de veces que su asistencia ha sido registrada mediante Bluetooth, y un listado de las fechas en las que estuvo presente. De este modo, los estudiantes pueden consultar su asistencia en cualquier momento sin necesidad de conexión a internet ni autenticar su identidad.



Figura 19: Lista de asistencia del propio alumno a una asignatura.

### ➤ Enviar preguntas al profesor

En cualquier momento durante la clase, el alumno puede seleccionar el dispositivo del profesor de su lista de asignaturas y enviar un mensaje o pregunta [Figura 20]. El módulo del profesor debe estar dentro de la

actividad encargada de recibir estos mensajes, y de ser así, el estudiante recibirá una confirmación de recibo. Tanto si el alumno no está registrado en la asignatura o su identificación personal es incorrecta, como si se ha producido algún error en la transmisión del mensaje, también recibirá en pantalla un mensaje con la información relativa al error.



Figura 20: Pantalla para enviar un mensaje al profesor.

### 3.3.3. Módulo de conexiones Bluetooth

Este módulo es el puente que conecta las actividades de la aplicación con el adaptador Bluetooth de cada dispositivo en el que se utilice la aplicación. En concreto, se encarga de crear los hilos de ejecución para crear las conexiones entre los sockets, definiéndolo como receptor en el caso del profesor, o como emisor en el caso de los alumnos. Además de conectar los sockets, se encarga de transmitir los mensajes desde la actividad hasta el *stream* de salida para que sean enviados mediante el adaptador Bluetooth, así como de leer los mensajes recibidos y enviarlos al *handler* de cada aplicación que lo solicite para interpretarlos.

Durante el desarrollo del proyecto nos encontramos con una serie de problemas que fuimos solventando según aparecían. En primer lugar, para realizar un intercambio de datos vía Bluetooth es necesario emparejar los dispositivos. Por este motivo decidimos separar los procedimientos de emparejamiento de dispositivos y registro del alumno. Es necesario emparejar los dispositivos en primer lugar antes de proceder a realizar una conexión en la que se tenga acceso a la dirección MAC del dispositivo del alumno. Este proceso puede realizarse también desde el menú de ajustes de Bluetooth, pero decidimos implementarlo dentro de la aplicación para que su uso sea más sencillo e intuitivo, y evitar así que el usuario tenga que salir de la aplicación para acceder a este menú.

El funcionamiento de la aplicación se basa en el intercambio de cadenas de caracteres entre los dispositivos móviles de profesor y alumnos. Una vez emparejados, se transmiten distintas cadenas en ambas direcciones en el socket que conecta ambos

dispositivos: el DNI del alumno, el contenido del mensaje enviado al profesor, o los mensajes de confirmación o error como resultado del procedimiento. Para esta tarea utilizamos como ejemplo la aplicación *Bluetooth Chat* [20] que ofrece la web de Android como ejemplo de conectividad Bluetooth, que consiste en un chat de dos usuarios en el que los mensajes se transmiten mediante una conexión Bluetooth, sin necesidad de cables ni conexión a internet. Sin embargo, esta aplicación admite una única conexión, ya que es un chat de dos dispositivos en el que uno de ellos está configurado como receptor. Para nuestra aplicación AppTendance necesitamos varias conexiones simultáneas, y dejar en espera a los usuarios que deseen conectar con el dispositivo receptor (en nuestro caso el del profesor) cuando éste esté ocupado con otras conexiones. Éste fue uno de los mayores problemas a la hora de desarrollar la aplicación, y se resolvió utilizando un bucle que intente la conexión infinitas veces hasta que ésta sea aceptada, o hasta que el usuario cancele la petición saliendo de la actividad.

Además, como se ha explicado anteriormente en el punto 3.1.1., cada dispositivo configurado como receptor puede conectarse con hasta un máximo de siete dispositivos, cada uno de ellos con un identificador de conexión único UUID. Gracias a esta característica, sería posible que el dispositivo del profesor reciba información de hasta siete dispositivos simultáneamente, reduciendo la espera de los alumnos a la hora de intentar la conexión.

### 3.3.4. Módulo de manejo de base de datos

Este módulo contiene la clase denominada *adaptador de la base de datos*. En esta clase se incluyen las funciones de apertura y clausura de la base de datos SQLite interna del dispositivo, así como las funciones encargadas de crear y declarar las tablas que se utilizarán en la aplicación, y que se describen en el punto 3.5.

Además este módulo incluye las funciones que sirven de acceso a la base de datos y que intercambian información con esta, bien en forma de inserciones, actualizaciones de datos existentes, eliminando filas, o realizando consultas. Entre estas funciones están, por ejemplo, las que ejecutan comandos SQL sobre la base de datos para editar, añadir o borrar la información relativa a los alumnos y las clases, así como los datos de las listas de asistencia. Este módulo también contiene subrutinas para actualizar la tabla de registro de alumnos donde se guardan y actualizan las direcciones MAC de los dispositivos de cada estudiante en el caso de que la aplicación se utilice por el profesor, y las direcciones MAC de los profesores de las asignaturas en el caso de que se utilice por un alumno. Por último, esta clase contiene una serie de métodos que ejecutan consultas SQL sobre la base de datos, devolviendo los datos resultantes de la consulta (información de alumnos, asistencia, comprobaciones de registro de direcciones MAC, etc.) en forma de “*cursor*”, es decir, una lista con la información de cada fila del resultado de la consulta.

### 3.3.5. Módulo de manejo de ficheros CSV

La aplicación desarrollada ofrece la opción de importar la información de los alumnos o exportar las listas de asistencia utilizando archivos en formato CSV (del

inglés *comma-separated values*) en la memoria externa del dispositivo. Este módulo contiene las funciones encargadas tanto de leer e interpretar el contenido de estos archivos y agregar los datos a la base de datos de alumnos, como de convertir las listas de asistencia almacenadas en la base de datos de la aplicación en un archivo CSV. Para ello, la aplicación crea una carpeta en el directorio raíz de la memoria externa del dispositivo denominada “AppTendance”, donde el usuario (en este caso el profesor) debe copiar sus archivos CSV con la información de los alumnos separada por punto y coma tal como sigue: “DNI; Nombre; Apellidos”. Igualmente, el profesor tiene la opción de exportar listas de asistencia globales de una asignatura, de forma que se creará en esta misma carpeta un archivo con un nombre compuesto por el nombre o identificador de la asignatura seguido de las fechas primera y última en las que se ha recogido asistencia. El formato de este fichero exportado será: “Apellidos; Nombre; DNI; fecha; presente/ausente”, ordenados por fecha de más antiguo a más actual. De esta forma el profesor puede intercambiar información con otros dispositivos simplemente copiando estos archivos, y puede almacenar todas las listas de forma fácil en una memoria externa, o editar los datos con cualquier programa de edición de hojas de cálculo.

### **3.4. Descripción de la base de datos**

Los dispositivos Android cuentan con una base de datos relacional interna gestionada por SQLite. A diferencia de los sistemas de gestión de bases de datos cliente-servidor, el motor de SQLite [36] no es un proceso independiente con el que el programa se comunica, sino que la biblioteca de SQL se enlaza con la aplicación y pasa a formar parte integral de la misma. La aplicación utiliza la funcionalidad de SQLite a través de llamadas simples a las subrutinas y funciones descritas en el módulo de manejo de la base de datos (ver 3.4.4), de forma que se reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos. En su versión 3, SQLite permite bases de datos de hasta 2 Terabytes de tamaño, más que suficiente para almacenar todas las listas de alumnos y asistencia necesarias a lo largo de un curso académico.

La base de datos empleada para la aplicación desarrollada cuenta con 6 tablas que almacenan toda la información necesaria tanto si el usuario es un profesor como si es un alumno, o si el usuario la utiliza de las dos formas [Tabla 3].

TABLA DE ALUMNOS				
<u>DNI</u>	Nombre	Apellidos	MAC del dispositivo del alumno	
TABLA DE ASIGNATURAS CREADAS POR EL PROFESOR				
<u>Identificador de asignatura</u>		Descripción		
TABLA DE ASIGNATURAS REGISTRADAS POR EL ALUMNO				
<u>Identificador de asignatura</u>		MAC del dispositivo del profesor		
TABLA DE ASIGNATURAS Y ALUMNOS				
<u>Identificador</u>	Identificador de asignatura		DNI del alumno	
TABLA DE ASISTENCIA A LAS CLASES CREADAS				
<u>Identificador</u>	Fecha	Identificador de asignatura	DNI del alumno	Estado (Presente / Ausencia)
TABLA DE ASISTENCIA PROPIA DEL ALUMNO				
<u>Identificador</u>	Fecha		Identificador de asignatura	

Tabla 3: Detalle de las tablas que componen la base de datos de la aplicación.

### 3.4.1. Tabla de alumnos

En esta tabla se almacena la información personal de los alumnos que asistirán a las clases impartidas por el profesor que utiliza la aplicación. En la versión actual, sólo se almacenan los datos básicos: DNI, nombre y apellidos; pero en futuras mejoras se podrían incorporar más datos como dirección de correo electrónico, número de teléfono, notas de exámenes, foto, etc. Además hay un campo extra en esta tabla que almacena el número de identificación MAC del dispositivo utilizado por el alumno cuando se registró en la aplicación. De esta forma, cuando el alumno quiera conectar con el dispositivo del profesor vía Bluetooth, la aplicación podrá relacionar el dispositivo empleado para la comunicación con la identidad del alumno, evitando posibles fraudes a la hora de pasar lista.

### 3.4.2. Tabla de asignaturas creadas

Esta tabla almacena todas las asignaturas creadas por el profesor, con un identificador único para cada una de ellas, y una descripción de ayuda en caso de que el profesor imparta varias clases o grupos de la misma asignatura.

### 3.4.3. Tabla de asignaturas registradas

De forma similar al funcionamiento de la tabla que almacena la información de los alumnos en el dispositivo del profesor, la aplicación utiliza una tabla para almacenar



las asignaturas en las que el alumno está registrado, en la que uno de los campos es la identificación MAC del dispositivo del profesor que imparte la asignatura. Así, cuando el alumno quiera establecer una conexión Bluetooth con su profesor, no tendrá que seleccionar el nombre del dispositivo, sino el nombre de la asignatura en la que está presente, y la aplicación se encargará de buscar en esta tabla la dirección del dispositivo con el que se quiere conectar.

#### **3.4.4. Tabla de alumnos registrados**

En esta tabla se relaciona el DNI de cada uno de los alumnos con el identificador de cada una de las asignaturas en las que éste alumno está registrado. A la hora de pasar lista, la aplicación comprueba en esta tabla que el alumno que intenta establecer la comunicación Bluetooth con el profesor está registrado en la asignatura.

#### **3.4.5. Tabla de asistencia**

Es la tabla que almacena más información. En ella se almacenan todas las listas de asistencia de todas las asignaturas impartidas por el profesor. Los campos de los que se compone esta tabla son: la fecha de la clase, el DNI del alumno, el identificador de la asignatura, y si el alumno estuvo ausente o presente. De esta forma, con una simple consulta a la base de datos, podemos filtrar las listas de asistencia por fecha, asignatura o alumno concreto.

#### **3.4.6. Tabla de asistencia propia**

Uno de los objetivos de la aplicación desarrollada es que el alumno esté informado del proceso de asistencia en todo momento, por lo que es necesario almacenar los datos y estadísticas de asistencia del propio alumno en su dispositivo. Por esto, cada vez que el alumno confirma su asistencia por medio de Bluetooth, se anota en esta tabla la fecha, de forma que el alumno pueda en cualquier momento consultar el número de veces y en qué fechas ha asistido a clase.

### **3.5. Descripción de las conexiones Bluetooth**

Como ya se ha comentado anteriormente, el funcionamiento de la aplicación desarrollada se basa en la comunicación inalámbrica por medio de Bluetooth. El mecanismo utilizado en la aplicación se basa en la aplicación de ejemplo de Google para comunicaciones inalámbricas denominada “Bluetooth Chat” [20]. Como vemos en esta aplicación de ejemplo, para que esta conexión sea posible, lo que hay que comprobar en primer lugar es que el dispositivo utilizado tenga un adaptador de Bluetooth, y que éste esté activado a la hora de entrar a una actividad que vaya a usarlo. Es por esto que al comenzar alguna de estas actividades, se comprueba el estado del adaptador Bluetooth, y en caso de que no se encuentre activado, se solicitará permiso al usuario para activarlo, saliendo de la actividad en caso de que no se conceda este permiso [Figura 21].

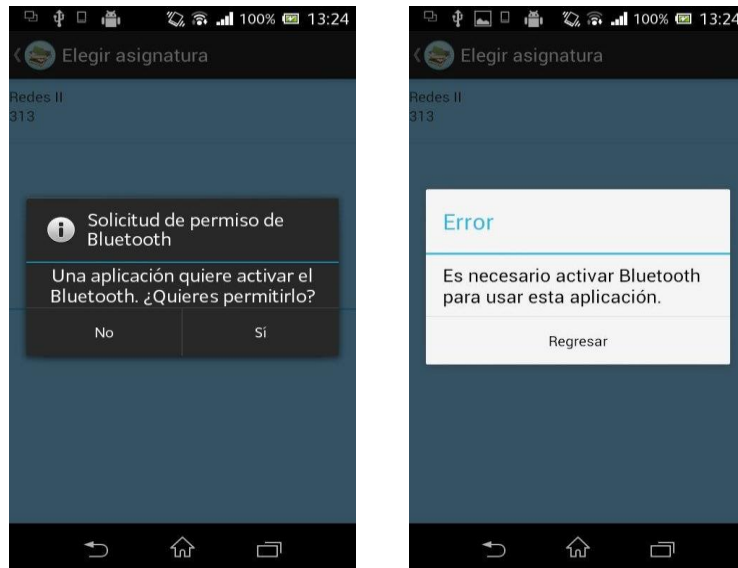


Figura 21: Petición para activar Bluetooth en el dispositivo y mensaje de error en caso de que el usuario no dé el permiso.

A partir de este momento, el adaptador Bluetooth está activo y listo para establecer comunicación con otro dispositivo. Pero antes de conectar e intercambiar información es necesario establecer un vínculo o enlace entre los dos dispositivos que se quieren conectar. Este enlace solicita el permiso de ambos usuarios para aceptar el vínculo, y, en caso de ser aceptado, se almacenará en la memoria del adaptador Bluetooth, de forma que no hará falta que el usuario vuelva a confirmar conexiones entrantes del dispositivo emparejado. En el caso de la aplicación desarrollada, la actividad que permite al alumno registrarse en una asignatura escanea y muestra en pantalla los dispositivos con Bluetooth activo y visibles en el entorno cercano. El descubrimiento de dispositivos consume muchos recursos, por lo que al terminar de escanear el entorno debemos detener la búsqueda. Para esta labor declaramos un *BroadcastReceiver* que recibe información por parte del adaptador Bluetooth cuando ha encontrado un nuevo dispositivo o bien cuando ya no encuentra más dispositivos, y los filtros necesarios para que sólo reciba estas dos acciones.

Al seleccionar alguno de los dispositivos cercanos no emparejados se enviará una solicitud de emparejamiento [Figura 22].

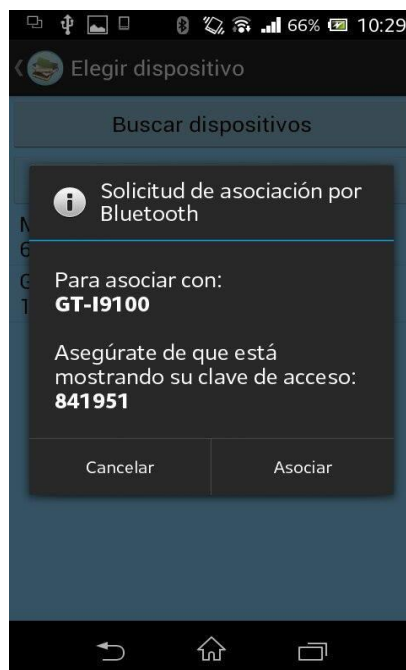


Figura 22: Solicitud de asociación por Bluetooth con otro dispositivo.

Desde el momento en el que ambos usuarios acepten el emparejamiento se pueden establecer conexiones Bluetooth en cualquier momento sin pedir confirmación previa, siempre que se disponga del identificador del otro dispositivo. Para realizar un intercambio de información entre dos dispositivos, es necesario configurar el adaptador de uno de ellos como receptor. Para ello, dentro del módulo de control del Bluetooth se define un hilo de ejecución que configura el adaptador para que escuche conexiones entrantes con una clave única UUID (*Universally Unique Identifier*). Esta clave UUID es generada aleatoriamente a la hora de programar la aplicación, pero es importante que los otros dispositivos que quieran conectarse configuren su adaptador Bluetooth para establecer la conexión con esta misma clave.

Al lanzar estos dos hilos en dos dispositivos emparejados, se establece un Socket que conecta ambas aplicaciones. En este socket escribiremos la información que deseemos intercambiar entre los dispositivos, de forma que el módulo encargado de controlar el adaptador Bluetooth realizará acciones sobre el Socket, y enviará los resultados a los *Handlers* de cada actividad. En el caso de que el alumno quiera registrar su dispositivo a la hora de registrar una nueva asignatura, escribirá en el Socket su DNI, y el dispositivo del profesor leerá este mensaje y lo enviará al *Handler* para que compruebe que el DNI es correcto y lo almacene en la base de datos de la asignatura, tras lo cual escribirá en el mismo socket un mensaje de confirmación, que leerá el dispositivo del alumno, y enviará al *Handler* indicado en su caso para que proceda a almacenar la asignatura en la base de datos de asignaturas registradas.

También es importante que los usuarios estén informados en todo momento de los incidentes ocurridos durante la comunicación. De esta tarea también se encargan los *Handlers* de cada actividad, mostrando un *Toast* o una alerta en caso de error en la comunicación.

Además, a la hora de pasar lista, varias decenas de dispositivos intentarán conectarse al mismo tiempo, por lo que establecemos un mecanismo de espera mediante el cual si un dispositivo no consigue conectarse, cancela el Socket de comunicación y lo vuelve a intentar crear, hasta que o bien se establezca una comunicación, o bien el usuario cancele el proceso. Por esto mismo, la aplicación, al recibir una conexión entrante, intenta hacer el mínimo trabajo posible antes de enviar el mensaje de confirmación y cerrar esta comunicación para recibir una nueva en el menor tiempo posible.

## 4. Evaluación del sistema

---

En el desarrollo de AppTendance, se han realizado una serie de pruebas que ratifican el correcto funcionamiento de cada uno de los módulos. Estas pruebas se dividen en pruebas de navegación por los distintos menus de la aplicación, pruebas de persistencia de la base de datos tanto a la hora de modificar la información de los alumnos o las clases como al anotar la asistencia, pruebas en las que se importan y exportan ficheros CSV, y pruebas de las conexiones Bluetooth para emparejar y registrar dispositivos y para intercambiar información

### 4.1. Pruebas de uso

Durante el desarrollo de la aplicación se han realizado pruebas tanto en dispositivos físicos como en el emulador GenyMotion. Sin embargo, dado que la aplicación requiere comunicaciones Bluetooth que no pueden ser simuladas por el emulador, ha sido necesario utilizar varios dispositivos físicos para probar todas las funcionalidades de la aplicación. Las pruebas realizadas aseguran el correcto funcionamiento de la navegación entre actividades, la consistencia de la base de datos y su correcto manejo por parte de la aplicación a la hora de modificar la información almacenada, y por último, de la comunicación Bluetooth entre dispositivos.

#### 4.1.1. Pruebas de navegación

Utilizando el emulador GenyMotion para simular varios dispositivos con varias versiones Android a partir del nivel de API 7 se ha probado la navegación de la aplicación, enlazando de forma correcta en todos los casos las actividades, y evitando el acceso accidental desde el módulo del profesor al del alumno y viceversa.

#### 4.1.2. Pruebas de la base de datos

Para comprobar que la base de datos funciona correctamente y la información es almacenada y editada de forma correcta, se realizaron las siguientes pruebas:

- **Crear / Editar / Borrar alumnos de una asignatura**

En el menú de edición de asignaturas, realizamos varias pruebas con los alumnos registrados en cada una de las asignaturas, comprobando que la información se modificaba correctamente y se mantenía persistente durante el uso de la aplicación.

- **Crear / Editar / Borrar alumnos de la lista general de alumnos registrados**

De igual forma, comprobamos que al modificar el listado general de alumnos, la información se veía reflejada en las listas de alumnos registrados y en las listas de asistencia de las asignaturas, evitando la repetición de DNI o de identificadores de asignaturas.

- **Crear / Editar / Borrar asignaturas**

Al eliminar asignaturas, comprobamos que la información relativa a los alumnos de esta asignatura, así como las listas de asistencia, son también eliminadas.

- **Cargar datos de un fichero CSV**

Además de las pruebas realizadas para el módulo encargado de importar y exportar ficheros CSV, comprobamos que la información leída al importar se carga en la base de datos de forma correcta, impidiendo guardar dos alumnos con el mismo DNI.

- **Anotar asistencia de forma manual**

También se realizaron pruebas a la hora de pasar lista de forma manual, comprobando que las listas de asistencia son correctas.

- **Comprobar las listas de asistencia con todos los filtros**

Probamos a consultar las listas de asistencia de todas las asignaturas y de todos los alumnos, mostrando mensajes de error al seleccionar fechas en las que no se pasó lista, o alumnos que no han asistido a ninguna clase. Además podemos comprobar que los resultados estadísticos de asistencia son correctos siempre.

- **Modificar las listas de asistencia**

Tanto al pasar lista de forma manual como mediante Bluetooth, probamos a modificar la asistencia tomada en ese momento, de forma que pudimos comprobar que la base de datos y las listas de asistencia se actualizan correctamente. También se realizaron pruebas modificando las listas de días pasados, con resultados igual de satisfactorios.

### **4.1.3. Pruebas de importación y exportación de ficheros CSV**

Para la realización de estas pruebas, intentamos importar los datos de los alumnos de una asignatura desde varios tipos y formatos de fichero, proporcionando siempre errores de formato (y el número de línea en el que se produce el error) en los casos que así se requiere, y guardando en la base de datos la información del fichero en el caso de que el formato de este sea correcto.

También se comprobó que la información de las listas de asistencia contenida en el fichero exportado, así como su formato, fueran correctos. Además, comprobamos que se pudiera sacar estos archivos de la memoria del dispositivo a un ordenador, y abrir estos ficheros con un editor de hojas de cálculo (Excel).

### **4.1.4. Pruebas de las conexiones Bluetooth**

A la hora de comprobar que el intercambio de información entre dispositivos por medio del Bluetooth se realizaba correctamente fue necesario crear una batería de pruebas que abarcara la mayor parte posible de los casos y escenarios.

En primer lugar, fue necesario comprobar que tanto el emparejamiento como el registro de la dirección MAC del alumno se realizaran de forma correcta, de forma que al desemparejar el dispositivo desde los ajustes del teléfono, éste desapareciera de la lista de dispositivos disponibles del alumno. Además, al intentar utilizar un dispositivo con una dirección MAC distinta a la del registro inicial para conectarse al dispositivo del profesor mediante la aplicación desarrollada se obtiene un mensaje de error.

También se probó que el proceso de la confirmación de la asistencia funcionara de forma correcta, obteniendo los mensajes de error indicados para fallos en el código de confirmación, asistencias confirmadas por segunda vez, o errores en la identificación personal del alumno o en la de su dispositivo. De la misma forma se realizaron pruebas sobre la funcionalidad de envío y recepción de mensajes entre profesor y alumno, obteniendo resultados correctos, y las alertas sonoras esperadas en el caso del dispositivo del profesor.

Sin embargo, lo más difícil de probar fue el funcionamiento de la aplicación con varios dispositivos intentando conectarse al mismo tiempo y desde distintos rangos de distancia, dado que no disponíamos de un número elevado de dispositivos Android para hacer las pruebas. A pesar de este inconveniente, las pruebas iniciales se llevaron a cabo realizando sólo una conexión repetidas veces, demostrando que la aplicación abría y cerraba los Sockets correctamente, liberando las conexiones cuya información había terminado de ser intercambiada.

## **4.2. Caso de uso general**

En este apartado describiremos un caso de uso general de la aplicación, en el que, paso por paso, mostraremos la secuencia de acciones que debe seguir el profesor para crear una asignatura, pasar lista, recibir mensajes durante la clase y consultar las

listas de asistencia, y los alumnos para confirmar su asistencia y enviar mensajes al profesor.

#### 4.2.1. Profesor

A continuación se presentan los casos de uso en el caso de que el usuario sea un profesor y desee crear una asignatura, cargar los datos de los alumnos de un fichero CSV, registrar a sus alumnos y pasar lista.

##### 4.2.1.1. Crear asignatura

En primer lugar, el profesor debe crear una asignatura y añadir los datos de los estudiantes que estén matriculados en ésta. Para ello debe entrar en el apartado de “Alumnos y Asignaturas” -> “Asignaturas” -> “Nueva Asignatura”, rellenar los campos con la información de la nueva asignatura, y proceder a ingresar la información de los estudiantes; en este caso de uso [Figura 23], mediante la carga de un fichero en formato CSV que previamente hemos copiado en la memoria externa del dispositivo y que está incluido en los Anexos de la presente memoria.

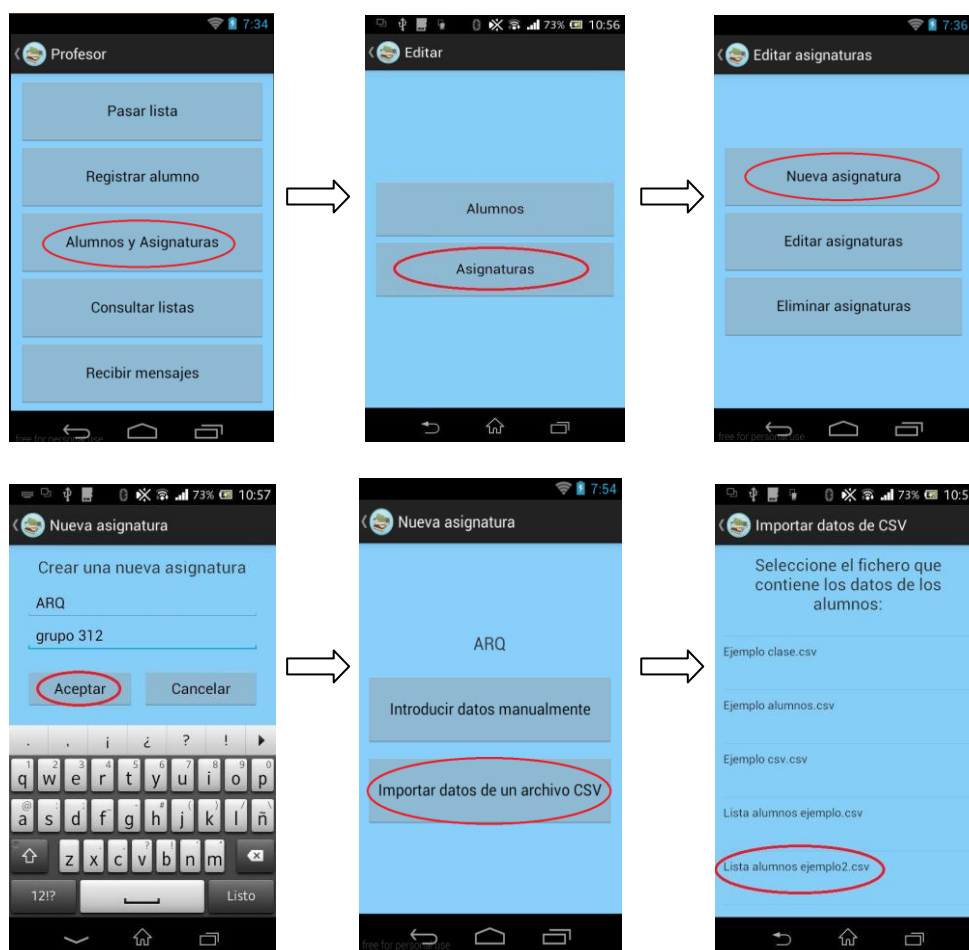


Figura 23: Caso de uso en el que el usuario crea una asignatura y carga los datos desde un fichero CSV.



#### 4.2.1.2. Registrar dispositivos de los alumnos

Una vez cargados los datos de los alumnos en la nueva asignatura, el profesor ha de registrar los dispositivos de los alumnos para almacenar la dirección MAC de cada uno de ellos, y a la vez enviar su propia dirección MAC para que los alumnos la almacenen en su aplicación y puedan usarla al confirmar su asistencia [Figura 24].

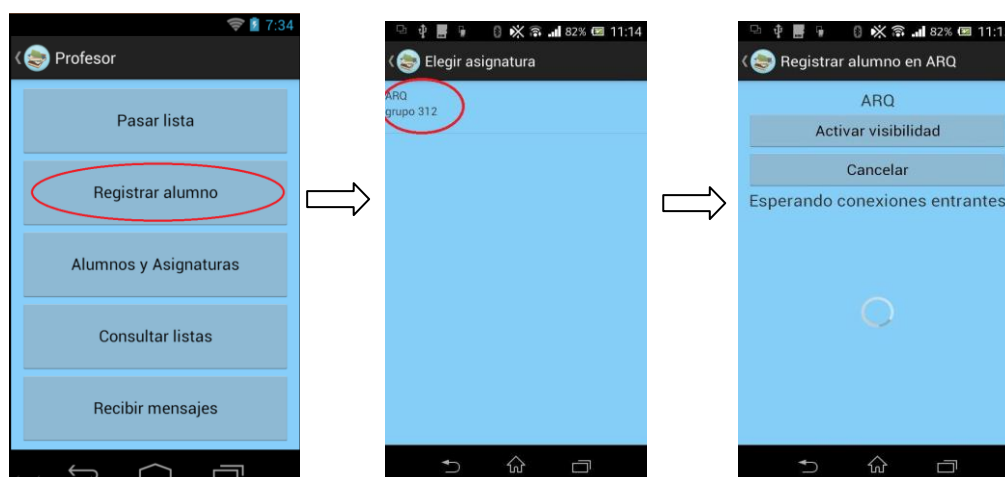


Figura 24: Caso de uso en el que el usuario registra los dispositivos de los alumnos.

En este punto la aplicación está a la espera de conexiones entrantes, por lo que el usuario puede seleccionar la opción de “Activar visibilidad” para hacer que su dispositivo Bluetooth [Figura 25] sea visible y pueda ser encontrado en los escaneos de los demás dispositivos y éstos puedan realizar el emparejamiento (ver punto 4.2.2.1) antes de realizar el registro.

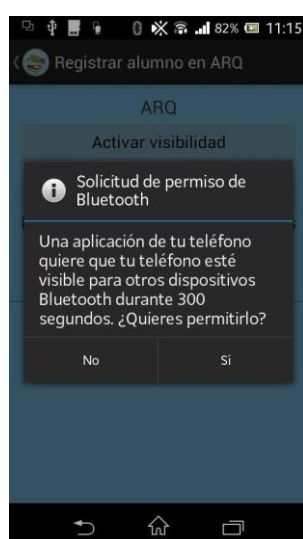


Figura 25: Solicitud de permiso de Bluetooth para hacer el dispositivo visible.

En este punto el alumno debe registrar su dispositivo (ver punto 4.2.2.2), tras lo cual el dispositivo del profesor mostrará un mensaje de confirmación o de error [Figura 26].

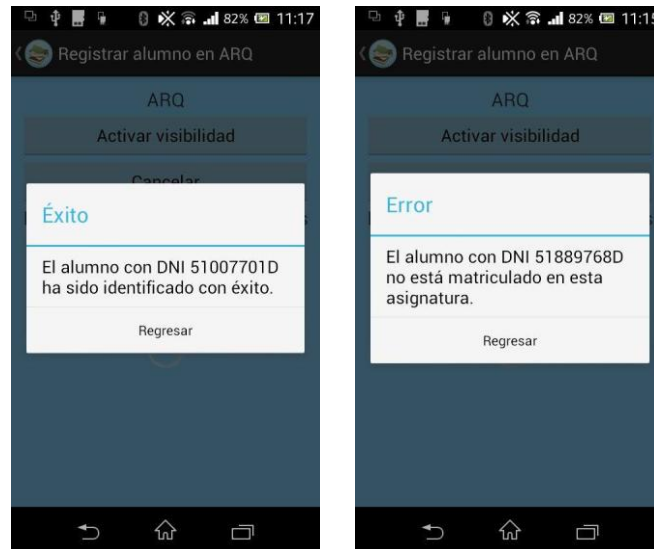


Figura 26: Mensajes de confirmación y error tras registrar un alumno.

#### 4.2.1.3. Pasar lista

Una vez el profesor ha creado la asignatura, añadido los datos de los alumnos matriculados en ella y registrado sus dispositivos móviles en la aplicación, el profesor puede proceder a pasar lista, bien de forma manual o bien a través de Bluetooth.

##### 4.2.1.3.1. Manualmente

Para pasar lista de forma manual, el profesor tendrá que, una vez seleccionada la asignatura, marcar con un tic todos los alumnos presentes y aceptar los cambios. Tras aceptar, se mostrará la lista de asistencia completa junto a una pequeña estadística de porcentajes de ausencia [Figura 27].

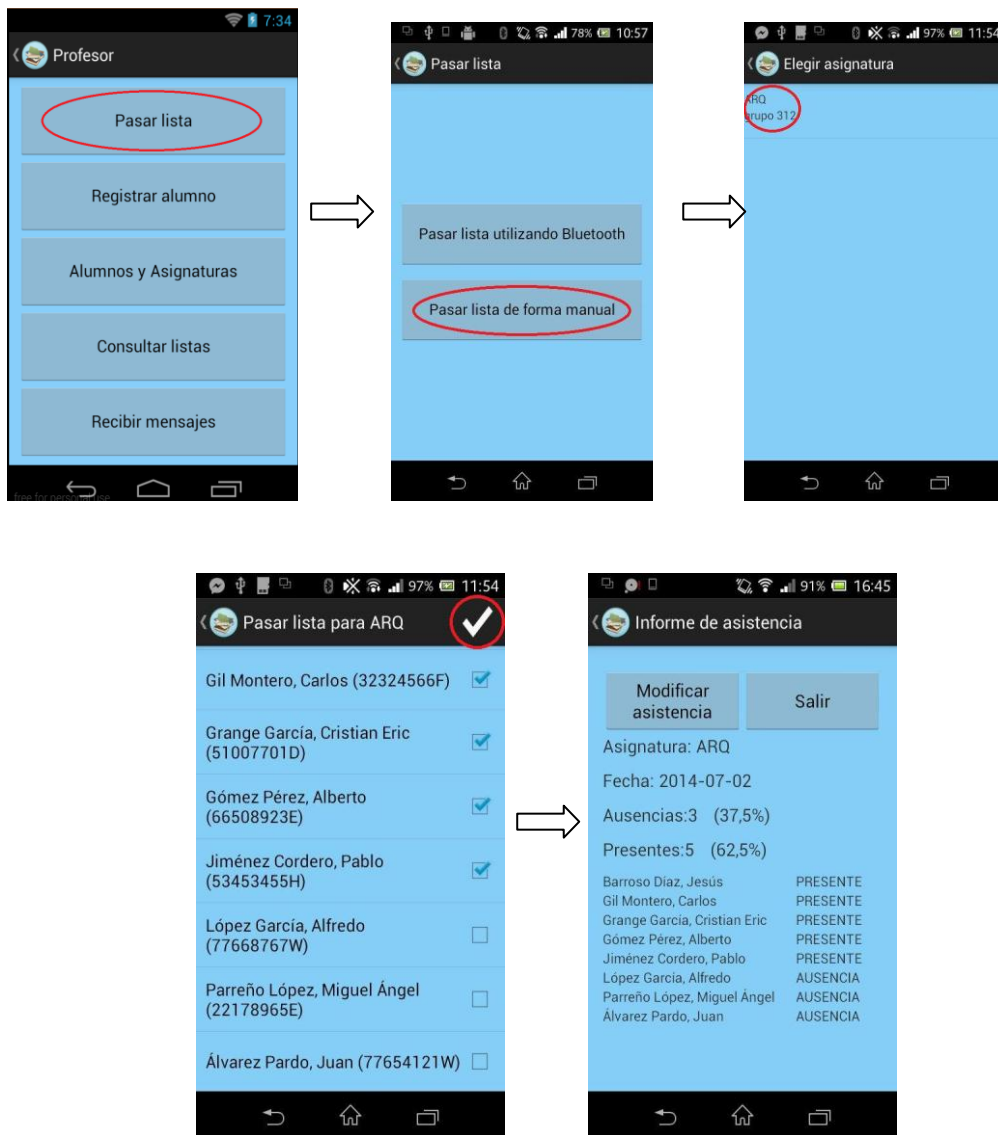


Figura 27: Caso de uso en el que el usuario pasa lista a su clase de forma manual.

#### 4.2.1.3.1. Mediante Bluetooth

Si el profesor desea anotar la asistencia de forma automática mediante Bluetooth debe escoger la asignatura e introducir el código de control que deben enviar sus alumnos para confirmar su asistencia a esa clase. A continuación el dispositivo se configura para recibir conexiones Bluetooth de los alumnos, y, por cada alumno que confirme su asistencia con éxito (ver el punto 4.2.2.3), se mostrará su nombre y su DNI en pantalla en el orden en el que se anote la asistencia. En el momento en el que el profesor decida parar el proceso, la asistencia se guardará en la base de datos interna del dispositivo y se mostrará por pantalla la lista de asistencia de ese día y la estadística, al igual que al pasar lista de forma manual [Figura 28].

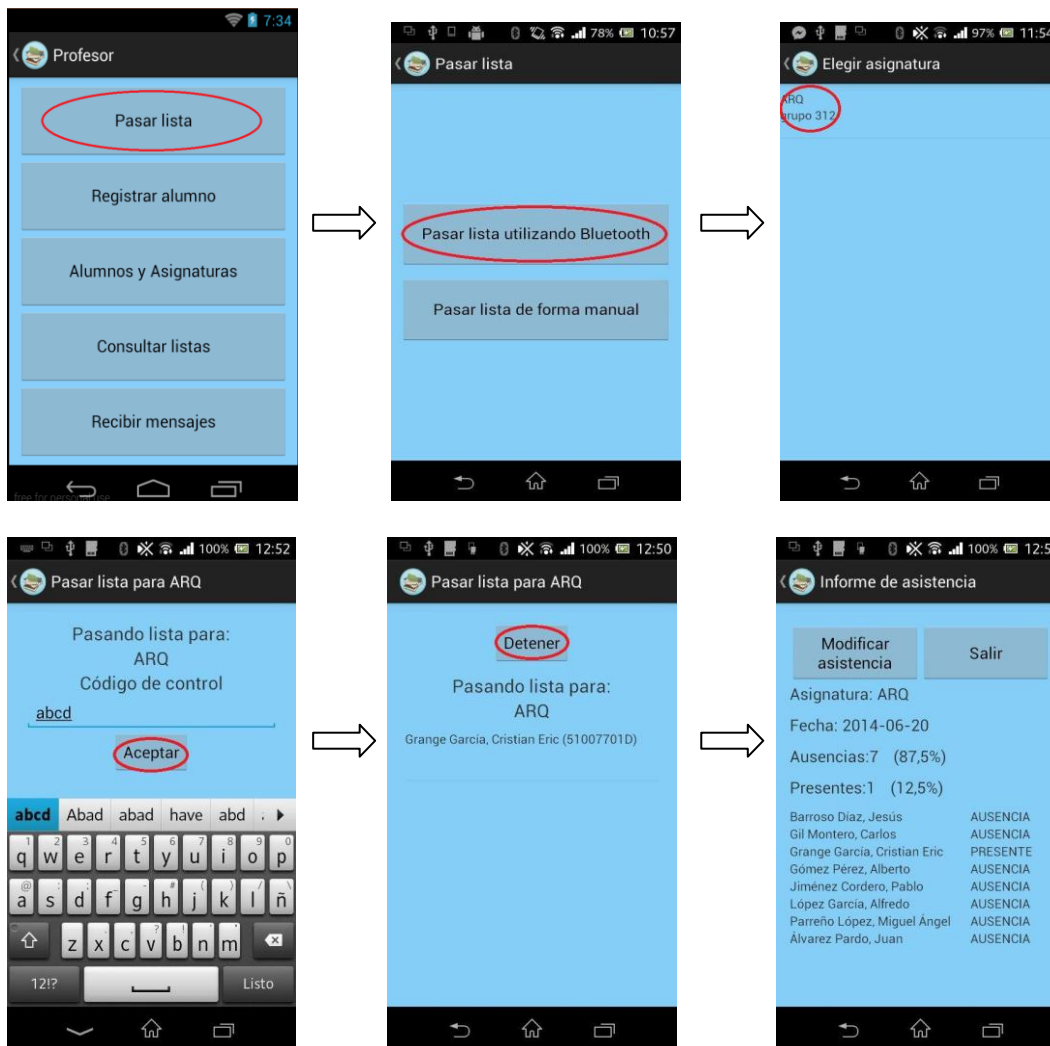


Figura 28: Caso de uso en el que el usuario pasa lista a su clase mediante Bluetooth.

#### 4.2.1.4. Recibir mensajes

En cualquier momento el profesor puede acceder a la actividad de la aplicación encargada de recibir mensajes de los alumnos presentes en la clase. Para ello, debe elegir la opción “Recibir mensajes” del menu principal y seguidamente seleccionar la asignatura deseada. Entonces AppTendance configurará el adaptador Bluetooth del dispositivo para que quede a la espera de conexiones entrantes [Figura 29]. En este momento, cualquier alumno puede realizar una conexión Bluetooth con el dispositivo del profesor y enviar un mensaje o pregunta, recibiendo en su dispositivo la confirmación del envío, o bien el error si lo hubiera (alumno no matriculado en la asignatura, error de conexión, etc.).

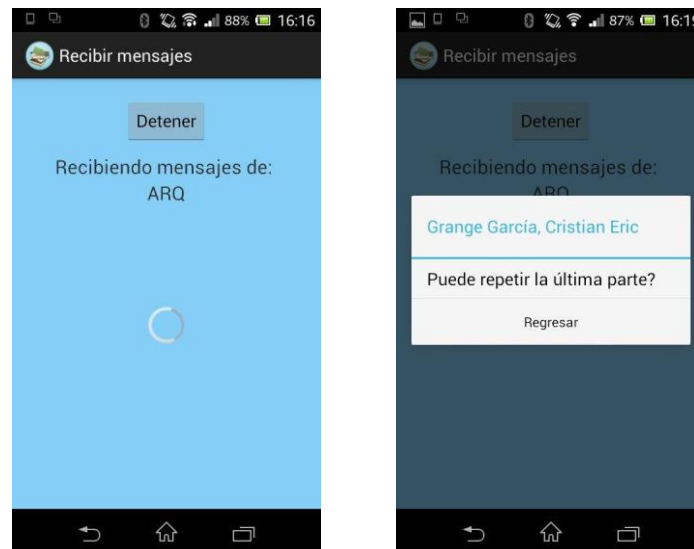


Figura 29: Recepción de mensajes de los alumnos.

#### 4.2.1.5. Consultar listas de asistencia

Para consultar las listas de asistencia almacenadas en el dispositivo hasta el momento basta con pulsar la opción de “Consultar listas”, tras lo que la aplicación nos dará a elegir la asignatura que queremos consultar, y después varias opciones de búsqueda: “Ver asistencia total”, “Ver asistencia por fecha” y “Ver asistencia por alumno” [Figura 30]. Además, el usuario también tiene la opción de “Modificar asistencia”, en la que podrá modificar la asistencia de cualquier alumno de la asignatura.

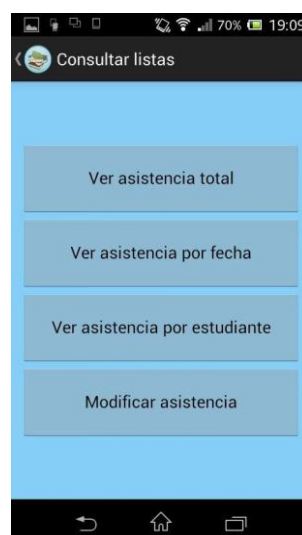


Figura 30: Opciones de búsqueda en listas de asistencia.

#### 4.2.1.5.1. Ver asistencia total

Al seleccionar esta opción, se muestra al usuario una serie de datos: las fechas primera y última de las que se tiene registrada asistencia en la aplicación, el número de ausencias y de alumnos presentes a lo largo de todas las clases en las que se recogió asistencia así como el porcentaje sobre el total, y finalmente una tabla con la asistencia detallada [Figura 31]. Además, el usuario tiene la opción de exportar la lista a un fichero en formato CSV en la memoria externa del dispositivo.



Figura 31: Informe de asistencia total de la asignatura.

#### 4.2.1.5.2. Ver asistencia por fecha

Si el usuario decide reducir la búsqueda a una fecha concreta, al seleccionar esta opción el usuario debe introducir la fecha sobre la que desea consultar la asistencia registrada. Después, si existe algún registro de asistencia de esa fecha en la base de datos del dispositivo, se muestra por pantalla un resumen similar al mostrado en el punto anterior, pero en el que solamente se mostrará la información relativa al día seleccionado [Figura 32].

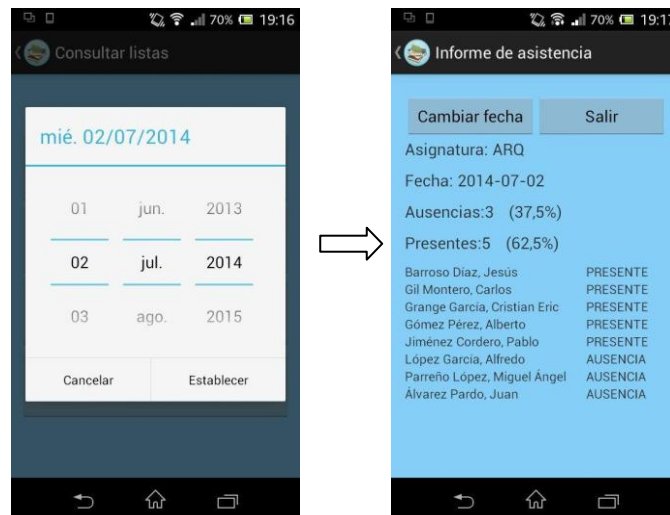


Figura 32: Lista de asistencia de la asignatura en una fecha concreta.

#### 4.2.1.5.3. Ver asistencia por alumno

Por último, el usuario puede consultar la asistencia completa de un único alumno de la clase, de forma que se mostrará un informe con el número de ausencias y asistencias del alumno en la asignatura con sus porcentajes correspondientes, y un listado de todas las fechas en las que se recogió asistencia para la asignatura y si el alumno estuvo presente o no [Figura 33].



Figura 33: Lista de asistencia de un alumno concreto de la asignatura.

### 4.2.2. Alumno

Siguiendo con el caso de uso cuya finalidad es crear una asignatura, registrar a los alumnos y, finalmente, pasar lista, en esta sección se describe el proceso que debe seguir el alumno: emparejar el dispositivo del profesor, registrar su dirección MAC y confirmar su asistencia.

#### 4.2.2.1. Emparejar dispositivo

Es el primer paso antes de iniciar cualquier conexión Bluetooth. Para ello el alumno debe seleccionar “Asignaturas” → “Nueva Asignatura” → “Emparejar un dispositivo”. A continuación, tras pulsar el botón de Buscar Dispositivos, se mostrará en pantalla una lista con todos los dispositivos con Bluetooth activo y visible en las cercanías de donde se encuentre el alumno. Debe seleccionar el dispositivo del profesor y que ambos acepten el emparejamiento en sus respectivos dispositivos [Figura 34].

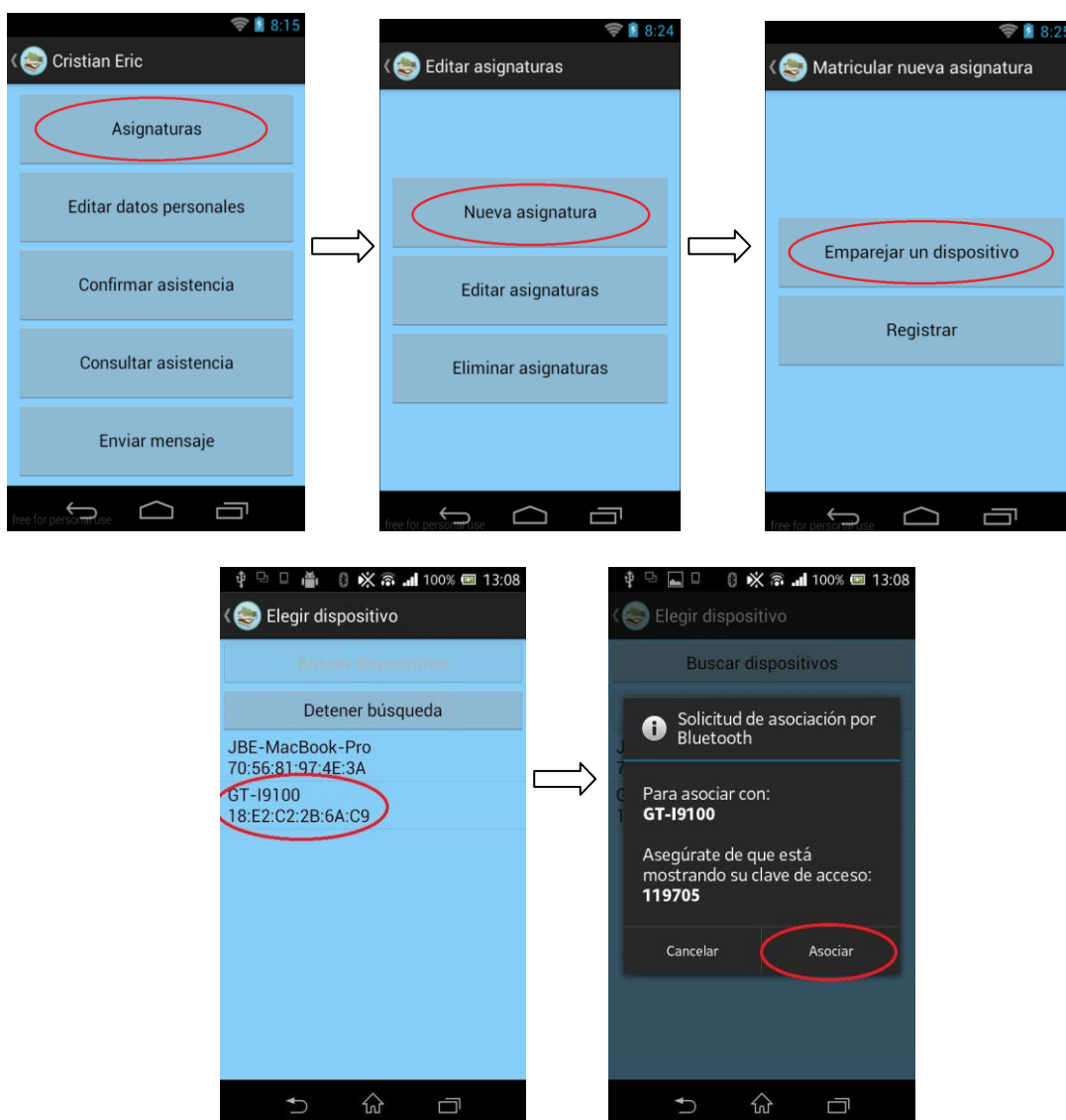


Figura 34: Caso de uso en el que el usuario empareja el dispositivo del profesor.



#### 4.2.2.2. Registrar dispositivo

Tras emparejar ambos dispositivos, el alumno debe registrar su dirección MAC única. Para ello, la aplicación mostrará una lista con los dispositivos con Bluetooth emparejados, de la que el usuario debe escoger el dispositivo del profesor de la asignatura que quiere registrar. Una vez establecida la conexión, se pedirá una confirmación para enviar el DNI, y, en caso de aceptar, el dispositivo del profesor recibirá este DNI, lo buscará en la base de datos de alumnos matriculados en la asignatura, y enviará un mensaje de confirmación, o de error en el caso de que el alumno no pertenezca a su clase. Si el registro se ha realizado correctamente, se pedirá al usuario que introduzca un nombre para la asignatura, al que se asociará automáticamente la dirección MAC del dispositivo del profesor para establecer las conexiones Bluetooth en el futuro [Figura 35].

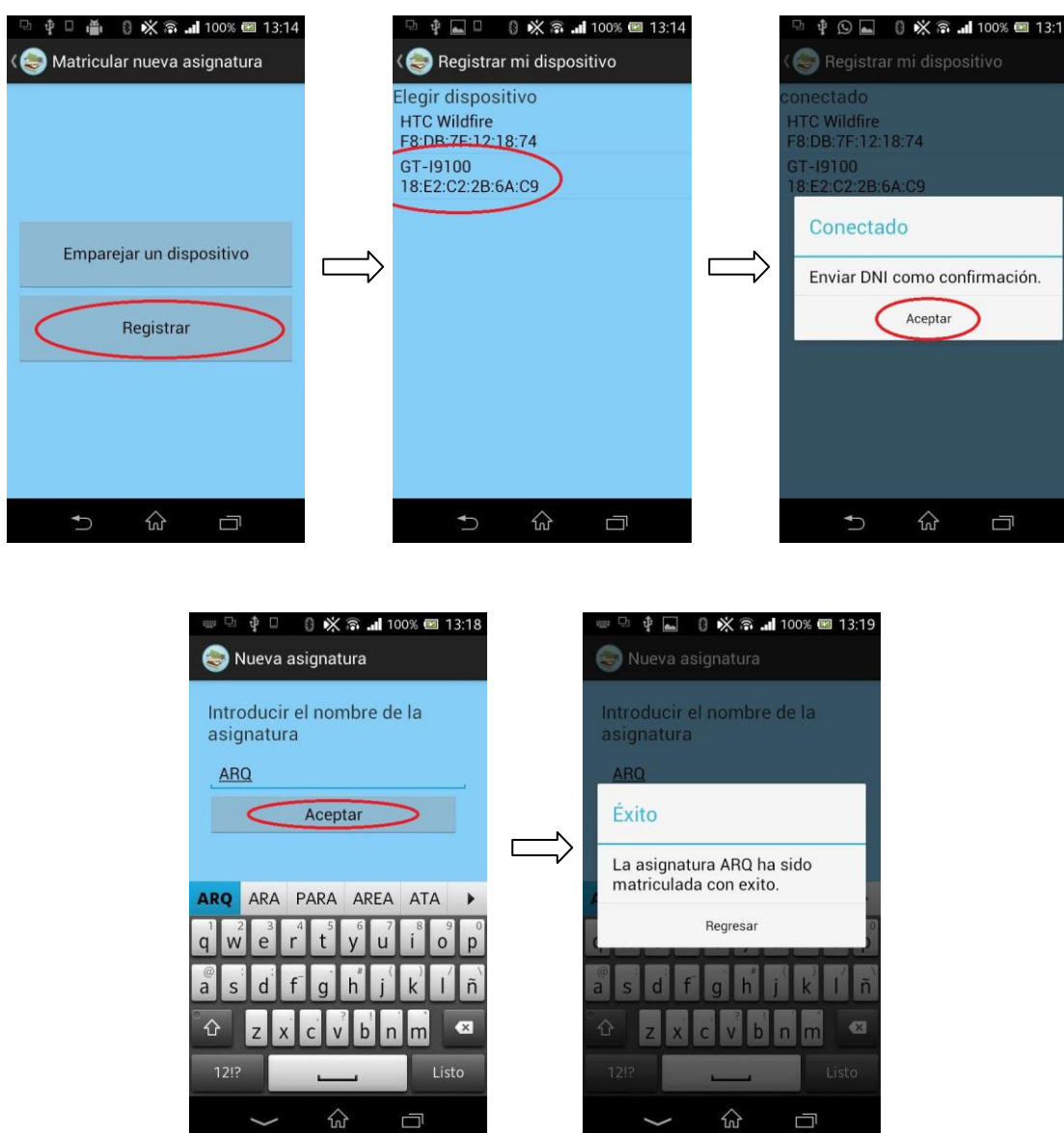


Figura 35: Caso de uso en el que el usuario registra su dispositivo añadiendo una nueva asignatura.

#### 4.2.2.3. Confirmar asistencia

Finalmente, tras emparejar y registrar el dispositivo la primera vez, el alumno puede confirmar su asistencia automáticamente mediante Bluetooth todas las veces que desee de aquí en adelante sin volver a realizar el registro. Para ello sólo tendrá que seleccionar la asignatura a la que está asistiendo, introducir el código de control proporcionado por el profesor y esperar a que se establezca la conexión, recibiendo un mensaje de confirmación o error [Figura 36].

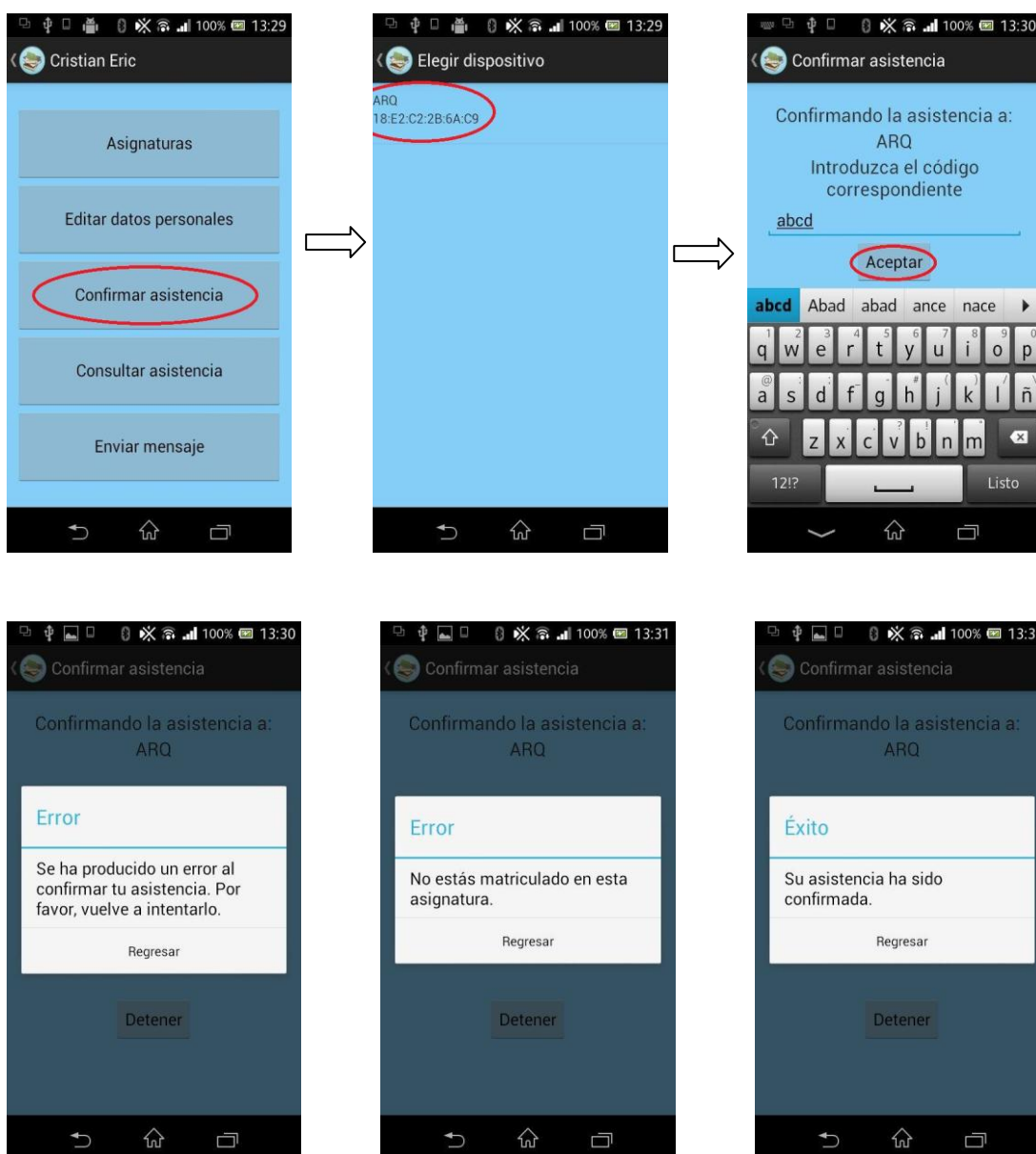


Figura 36: Caso de uso en el que el usuario confirma su asistencia a clase, y los mensajes de confirmación o error recibidos tras realizar el procedimiento.

#### 4.2.2.4. Enviar mensajes al profesor

En el transcurso de la clase, el alumno puede seleccionar la opción de “Enviar mensaje”. Después se le pedirá que seleccione la asignatura a cuyo profesor quiere enviar el mensaje y el contenido del mismo [Figura 37]. A continuación, AppTendace intentará establecer una comunicación Bluetooth con el dispositivo del profesor, y enviará el contenido del mensaje. Si el dispositivo del profesor está configurado para recibir mensajes en ese momento (ver punto 4.2.1.4.), se mostrará en su pantalla el contenido del mensaje y el nombre del alumno que lo envió. En caso de no ser así, se mostrará en la pantalla del alumno un mensaje de error en la conexión.

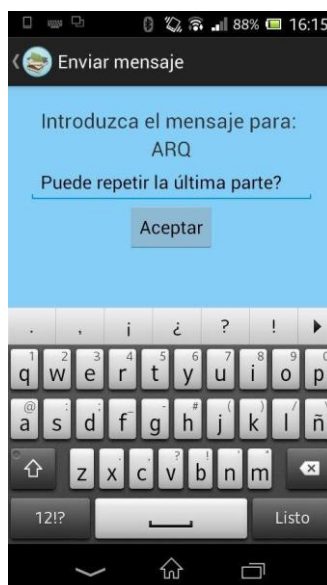


Figura 37: Caso de uso en el que el alumno envía un mensaje al profesor.

### 4.3. Pruebas en distintos modelos y versiones de Android

La aplicación se ha probado de manera satisfactoria en los siguientes modelos [Tabla 4]:

Modelo	Versión de Android	API	Resolución (píxeles)
Sony Xperia M	4.1.2	16	480x854
Samsung Galaxy S2 i9100	4.1.2	16	480x800

Tabla 4: Modelos de dispositivos móviles Android físicos en los que se ha probado la aplicación.

Además se ha simulado con GenyMotion en los siguientes dispositivos virtuales [Tabla 5]:

Modelo	Versión de Android	API	Resolución (píxeles)
Google Galaxy Nexus	4.2.2	17	720x1280
Google Nexus 10	4.3	18	2560x1600
HTC One	4.3	18	1080x1920
LG Optimus L3 II	4.1.1	16	240x320
Samsung Galaxy Note 3	4.4.2	19	1080x1920
Samsung Galaxy S5	4.4.2	19	1080x1920
Sony Xperia Tablet Z	4.2.2	17	1920x1200

Tabla 5: Modelos de dispositivos móviles Android simulados con GenyMotion.

Al probar la aplicación con el simulador, no es posible utilizar las conexiones Bluetooth, por lo que sólo se pudo probar el correcto funcionamiento de las actividades, los accesos a preferencias y base de datos, y la interfaz gráfica en todos los modelos y versiones de Android mostrados en la tabla.

#### 4.4. Pruebas con usuarios

Finalmente, se realizaron pruebas simulando un caso real de uso de la aplicación con hasta 3 dispositivos utilizando la aplicación como alumnos y uno como profesor. En primer lugar, se comprobó que el rango de distancia en el que se puede establecer una conexión Bluetooth es suficiente para que todos los alumnos de cualquier aula de la Escuela Politécnica Superior de la Universidad Autónoma de Madrid puedan utilizar la aplicación. También se comprobó que el ahorro de tiempo a la hora de pasar lista es considerable. Al pasar lista mediante Bluetooth, el dispositivo del profesor tarda unos 3 segundos para configurar el adaptador Bluetooth, y a continuación se recibe la asistencia de todos los alumnos de forma continua y, si todos los alumnos registran su asistencia a la vez, casi inmediata. En el caso de la prueba realizada, la asistencia de los 3 alumnos se registró por completo en 3 segundos, mientras que, en el caso de que el profesor desee pasar lista de forma manual, se necesitarían unos 4 o 5 segundos en leer el nombre del alumno, buscarlo e identificarlo visualmente y anotar su presencia en la lista, de forma que para 3 alumnos en una clase con mucha gente, se necesitarían más o menos 10 o 15 segundos, es decir, AppTendance es 5 veces más rápido.

## 5. Conclusiones y trabajo futuro

---

Con la llegada al mercado de los Smartphones, y posteriormente las tabletas, cada vez es más común y fácil automatizar tareas de la vida cotidiana de forma que éstas consuman menos tiempo y se realicen de un modo más sencillo y con menos esfuerzo. Es por esto que consideramos que el proceso de verificación de asistencia a clase podría ser automatizado y facilitar de esta forma la tarea de los profesores, así como ahorrar tiempo y preocupación a los alumnos que han de esperar su turno para confirmar su asistencia levantando el brazo o firmando en una hoja de papel que en ocasiones acaba perdiéndose.

Después de una detallada investigación de las tecnologías disponibles y los sistemas similares desarrollados hasta el momento, decidimos optar por la comunicación Bluetooth para transmitir información entre los dispositivos Android de alumnos y profesores, ya que se trata de tecnologías muy extendidas, y permiten un intercambio bidireccional de información, de forma que los alumnos reciban en el momento una confirmación de su verificación de asistencia, evitando así confusiones. La aplicación desarrollada permite, además, anotar la asistencia de forma manual, en caso de que alguno de los alumnos no disponga de ningún dispositivo móvil Android, o bien no lo tenga disponible en ese momento, por lo que nadie se ve excluido de utilizar este sistema. Al no requerir conexión a internet, ni ninguna instalación inicial de cámaras, lectores de tarjetas, chips de radiofrecuencia, etc. se convierte en una opción económica para cualquier profesor, bien en una universidad o bien en un pequeño grupo de estudio o trabajo. Además, al permitir al alumno enviar preguntas a su profesor durante el transcurso de la clase de forma que ningún otro alumno sepa quién realizó la pregunta, se pierde el miedo a ser juzgado por ello, y se reducirá, seguramente, el número de cuestiones sin responder en la cabeza de los estudiantes, mejorando la atención y el rendimiento.

Sin embargo, hay que tener en cuenta las limitaciones de esta aplicación. Dado que el protocolo de comunicación inalámbrica Bluetooth no asegura la presencia del dispositivo remoto a una distancia específica, sino que esta distancia es variable, no se puede asegurar que el alumno esté en el interior del aula a la hora de confirmar su asistencia. Por este motivo el profesor debe proporcionar un código de control a los alumnos, de forma que se asegure la presencia de éstos en el aula. Aunque este método no garantiza completamente la presencia del alumno en el aula, esto tampoco ocurre con los métodos tradicionales usados hasta ahora, por ejemplo cuando un alumno firma por otro, o dice “presente” en voz alta cuando el profesor dice el nombre de un alumno que no está en clase.

En la primera versión de la aplicación nos hemos centrado en la funcionalidad del sistema, pero en versiones futuras se trabajará en la interfaz de usuario para hacerla más atractiva y mejorar algunos aspectos de usabilidad. Además, se podría mejorar el formato de presentación de los informes de asistencia, incluyendo filtros de búsqueda dentro de cada documento. Para facilitar la labor de identificación de los alumnos, también se podría incluir información más detallada en la base de datos, con fotos, notas de las asignaturas, correo electrónico o número de teléfono, de modo que el profesor pueda ver rápidamente si algún alumno intenta engañarle al confirmar su asistencia, o

enviar un correo electrónico o SMS al alumno, si su ausentismo excede el máximo permitido en la asignatura. De la misma manera, en la versión actual de AppTendance es imposible para el alumno saber qué días no asistió a clase, por lo que cabe mencionar como trabajo futuro la posibilidad de ingresar un calendario de clases, o bien que el dispositivo del profesor intercambie los datos de asistencia a la hora del proceso de confirmación.

El sistema de control automático de asistencia crea un enlace entre profesor y alumnos que permite el intercambio de información en el momento, por lo que en futuras actualizaciones del sistema se podrían incorporar nuevas funcionalidades que aprovechen esta ventaja respecto a los otros sistemas desarrollados hasta el momento. Así, se podría trabajar en un módulo que realice, por ejemplo, preguntas o test en el transcurso de la clase, recogiendo las respuestas y almacenándolas en la base de datos del dispositivo del profesor, de forma que pueda ver los resultados de la atención que prestan sus alumnos en clase, y de lo bien que comprenden la asignatura, consultando informes y estadísticas de estos resultados. Cabe también mencionar la posibilidad de trabajar en una forma de automatizar el proceso de registro de los estudiantes en el dispositivo del profesor, como, por ejemplo, vincular la aplicación con *Sigm@*, o la red interna de cada institución, de forma que se carguen todos los datos a la vez: asignaturas matriculadas, información personal de los alumnos, etc. También se podría añadir la opción de importar listas de asistencia pasadas para, en el caso de que el profesor cambie de dispositivo, poder continuar registrando la asistencia de sus alumnos desde el punto en el que cambió de dispositivo.

En resumen, el sistema de control automático de asistencia ofrecido por AppTendance es una forma rápida, económica y viable de anotar y almacenar la asistencia de los alumnos a clase, que facilitará a los profesores o monitores de cualquier actividad tener actualizadas en todo momento sus listas, y sin temor a que éstas se pierdan. Dado el abanico de posibilidades descritas anteriormente, y la variedad de proyectos similares que están disponibles en el mercado, o funcionando en este momento en instituciones universitarias, el trabajo de desarrollo de esta aplicación no termina con la finalización de este proyecto, sino que puede continuar para facilitar aún más la labor del profesor.



## 6. Referencias

---

- [1] <http://www.eees.es/es/eees>. Consultado el 5 de Julio de 2014.
- [2] <http://www.eees.es/es/ects>. Consultado el 5 de Julio de 2014.
- [3] <http://www.nfcworld.com/2012/01/31/312879/spanish-students-to-register-for-classes-with-nfc/> (31 de Enero de 2012). Consultado el 5 de Julio de 2014.
- [4] <https://play.google.com/store/apps/details?id=com.aor.attendance> (29 de Marzo de 2014) . Consultado el 5 de Julio de 2014.
- [5] <https://play.google.com/store/apps/details?id=eu.innovaapps.mobileattendance> (11 de Abril de 2014). Consultado el 5 de Julio de 2014.
- [6] <https://play.google.com/store/apps/details?id=com.attendance.school> (29 de Junio de 2014). . Consultado el 5 de Julio de 2014.
- [7] <https://play.google.com/store/apps/details?id=com.proappdesigns.attendancemanager> (6 de Noviembre de 2013) . Consultado el 5 de Julio de 2014.
- [8] <http://www.irda.org/>. Consultado el 5 de Julio de 2014.
- [9] [http://es.wikipedia.org/wiki/Infrared\\_Data\\_Association](http://es.wikipedia.org/wiki/Infrared_Data_Association). Consultado el 5 de Julio de 2014.
- [10] Marcos J. López Fernández, Jorge G. Fernández, Sergio R. Aguilar, Blanca S. Selvi y Rubén G. Crespo, “*Control of attendance applied in higher education through mobile NFC technologies*”. In Expert Systems with Applications 40 (2013) 4478-4489.
- [11] S. Nainan, R. Parekh and T. Shah, “*RFID Technology Based Attendance Management System*”. IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 1, No 1, January 2013.
- [12] V. Bhalla, T. Singla, A. Gahlot and V. Gupta, “*Bluetooth Based Attendance Management System*”. In International Journal of Innovations in Engineering and Technology (IJET).
- [13] Frenzel, L. E. (2006), “*NFC makes great progress in the wireless world*”. Electronic design: For engineers and engineering managers 20, 36–37.
- [14] Javier Areitio Bertolín, “*Análisis de los riesgos y contramedidas en seguridad-privacidad de la tecnología NFC en móviles*”. Seguridad en Redes, Noviembre 2011, pg. 42-51.
- [15] S. Kadry and M. Smaili, “*Wireless attendance management system based on iris recognition*”. Scientific Research and Essays Vol. 5(12), pp. 1428-1435, 18 June, 2010.



- [16] S. K. Opoku, “*An Automated Biometric Attendance Management System with Dual Authentication Mechanism Based on Bluetooth and NFC Technologies*”. International Journal of Computer Science and Mobile Computing (IJCSMC), Vol. 2, Issue 3, March 2013, pg. 18-25.
- [17] D. Acharya and A. K. Mishra, “*Wireless Fingerprint Based Student Attendance System*”. Department of Electrical Engineering, National Institute of Technology Rourkela, 2010.
- [18] N. Kar, M. K. Debbarma, A. Saha and D. R. Pal, “*Study of Implementing Automated Attendance System Using Face Recognition Technique*”. International Journal of Computer and Communication Engineering, Vol. 1, No. 2, July 2012.
- [19] S. S. Chawhan, M. P. Girhale and G. Mankar, “*Mobile Phone Based Attendance*”. In IOSR Journal of Computer Engineering (IOSR-JCE) Volume 10, Issue 3 (Mar. – Apr. 2013), PP 48-50.
- [20] <http://developer.android.com/guide/topics/connectivity/bluetooth.html>. Consultado el 5 de Julio de 2014.
- [21] Justin Lawson Brown, “*Bluetooth Student Registration System*”.
- [22] Dr. S. R. Reddy, D. Goyal and A. Bansal, “*Mobile Based Attendance Management System*”.
- [23] <http://www.sqlite.org>. Consultado el 5 de Julio de 2014.
- [24] <http://www.bluetooth.com/Pages/Mobile-Telephony-Market.aspx>. Consultado el 5 de Julio de 2014.
- [25] <http://www.kantarworldpanel.com/global>. Consultado el 5 de Julio de 2014.
- [26] [www.bluetooth.com](http://www.bluetooth.com). Consultado el 5 de Julio de 2014.
- [27] William Stallings, “*Wireless Communications and Networks*”.
- [28] <http://developer.android.com/tools/sdk/eclipse-adt.html> (Junio de 2014) . Consultado el 5 de Julio de 2014.
- [29] <http://www.engadget.com/2010/10/01/google-expands-androidss-reach-accepting-paid-apps-from-20-mor/> (de 2011) . Consultado el 5 de Julio de 2014.
- [30] [www.ticbeat.com/sim/activan-700000-android-dia/](http://www.ticbeat.com/sim/activan-700000-android-dia/) (21 de Diciembre de 2011) . Consultado el 5 de Julio de 2014.
- [31] [www.kantarworldpanel.com/es/Noticias/Android-ya-est-en-9-de-cada-10-nuevos-smartphones](http://www.kantarworldpanel.com/es/Noticias/Android-ya-est-en-9-de-cada-10-nuevos-smartphones) (17 de Abril de 2013) . Consultado el 5 de Julio de 2014.

- [32] [http://es.wikipedia.org/wiki/Entorno\\_de\\_desarrollo\\_integrado](http://es.wikipedia.org/wiki/Entorno_de_desarrollo_integrado). Consultado el 5 de Julio de 2014.
- [33] <http://developer.android.com/sdk/installing/studio.html> (Junio de 2014) . Consultado el 5 de Julio de 2014.
- [34] <http://www.jetbrains.com/>. Consultado el 5 de Julio de 2014.
- [35] <http://www.genymotion.com/>. Consultado el 5 de Julio de 2014.
- [36] <http://es.wikipedia.org/wiki/SQLite>. Consultado el 5 de Julio de 2014.



## 7. Anexos

---

### 7.1. Fichero CSV de ejemplo

El fichero debe estar escrito en formato UTF-8 con el formato: DNI; NOMBRE; APELLIDOS, tal y como muestra el fichero de prueba “Lista alumnos ejemplo.csv”

```
51007701D;Cristian Eric;Grange García  
66508923E;Alberto;Gómez Pérez  
77654121W;Juan;Álvarez Pardo  
13454667F;Jesús;Barroso Díaz  
53453455H;Pablo;Jiménez Cordero  
32324566F;Carlos;Gil Montero  
22178965E;Miguel Ángel;Parreño López  
77668767W;Alfredo;López García
```